Lecture notes in Recursion Theory
Arnold W. Miller

These are lecture notes from Math 773. There were mostly written in 2004 but with some additions in 2007.

DESCRIPTION: Abstract theory of computation. Turing degree and jump, arithmetic hierarchy, index sets, simple and (hyper)hypersimple sets, Kleene-Post results in Turing degrees, finite injury priority arguments: Friedberg-Muchnik Theorem, Sacks Splitting Theorem, existence of a maximal set. Infinite injury priority arguments: Lachlan minimal pair, Sacks density theorem, Shoenfield incomplete high degrees. Recursive ordinals and the hyperarithmetical hierarchy.

Some general references in this area are:
Hartley Rogers, Theory of recursive functions, 1967
Robert Soare, Recursively enumerable sets and degrees, 1987
Piergiorgio Odifreddi, Classical recursion theory, vol 1,2 1989,1999
Barry Cooper, Computability theory, 2004
Robert Soare, Computability theory and applications, 2008

Contents

1 UR-Basic programming **3**

2 Primitive recursive functions **6**

3 Primitive recursive functions are UR-Basic computable **11**

4 UR-BASIC computable functions are recursive **12**

5 Church-Turing Thesis **16**

6 Universal partial recursive function **18**

7 The recursively enumerable sets **19**

8 Separation and reduction **23**

9 Many-one reducibility **24**

10 Rice's index Theorem **26**

11 Myhill's recursive permutation Theorem **27**

12 Roger's adequate listing Theorem **30**

13 Kleene's Recursion Theorem **31**

14 Myhill's characterization of creative set **33**

15 Simple sets **36**

16 Oracles **37**

17 Dekker deficiency set **37**

18 Turing degrees and jumps **38**

19 Kleene-Post: incomparable degrees **39**

20 The join **41**

21 Meets **42**

22 Spector: exact pairs **44**

23 Friedberg: jump inversion **46**

24 Spector: minimal degree **48**

25 Sacks: minimal upper bounds **50**

26 Friedberg-Muchnik Theorem **51**

27 Embedding in the r.e. degrees **54**

28 Limit Lemma and Ramsey Theory **56**

29 A low simple set **58**

30 Friedberg splitting Theorem **60**

31 Sacks splitting Theorem **62**

32 Lachlan and Yates: minimal pair **68**

33 Friedberg: A one-one enumeration of the r.e. sets **74**

34 Hypersimple sets **79**

35 Hyperhypersimple sets **85**

36 Maximal sets **88**

37 The lattice of r.e. sets **91**

38 Arithmetic hierarchy **99**

39 Post: Δ^0_2 same as recursive in $0'$ **101**

40 EMP, TOT, FIN, and REC **104**

41 Domination and high degrees **109**

42 High degrees using the Psuedojump **112**

43 First-order theories **117**

44 Analytic sets **120**

Appendix

45 Turing machines **130**

46 Trees, Konig's Lemma, Low basis **136**

1 UR-Basic programming

We begin by giving a formal definitions of computability, a toy programming language: UR-BASIC.

Variables are any string of letters or numerals, A-Za-z0-9.
Statements are of the form

Let $X = X + 1$
Let $X = X \dot{-} 1$
If $X \leq Y$ then goto k

where X and Y are any variables and k is a nonnegative integer, i.e. $k \in \omega$, which is a line number.

A UR-Basic program is a sequence S_0, S_1, S_2, …, S_n of statements. Variables only take on nonnegative integer values. The symbol $\dot{-}$ means subtraction unless the result is negative and then it yields zero. The program halts if we "goto" to a line $k > n$.

A function $f : \omega \to \omega$ is UR-Basic computable iff there exists a program P, designated input variable X and output variable Y such that for any $n \in \omega$ if we put $X = n$ and all other variables zero and start with the first statement of P, then P eventually halts with $f(n)$ in variable Y. There is a similar definition for $f : \omega^m \to \omega$ to be UR-Basic computable.

Next we indicate how to simulate more complex statements using these three kinds of statements. When substituting multiline statements for a single statement, the "goto" numbers must be adjusted.

Basic:	UR-Basic:
Go to k	If $X \leq X$ then goto k
Continue	Let Donothing=Donothing+1
Let Y=X	1 If $X \leq Y$ then go to 4
	2 Let Y=Y+1
	3 Go to 1
	4 If $Y \leq X$ then go to 7
	5 Let $Y = Y \dot{-} 1$
	6 Go to 4
	7 Continue
Constants	
0	this is a variable - we agree never to change it
1	let $1 = 1 + 1$
2	Let $2 = 2 + 1$
	Let $2 = 2 + 1$

If $X < Y$ then goto k	Let $tempX = X$ Let $tempX = tempX + 1$ if $tempX \leq Y$ then goto k
If $X = Y$ then goto k	1 If $X < Y$ then goto 4 2 If $Y < X$ then goto 4 3 Go to k 4 continue
For $i = 1$ to n S_1 ... S_k Next i	1 If $n = 0$ then goto 7 2 Let $i = 1$ 3 S_1 ... 4 S_k 5 Let $i = i + 1$ 6 If $i \leq n$ then goto 3 7 continue

Example 1.1 *The pair of functions remainder and quotient are UR-Basic computable i.e., input n, m then output q, r with $n = qm + r$ and $0 \leq r < m$.*

Proof
$n = qm + r$:

1 Let $q = 0$
2 Let $r = n$
3 If $r < m$ then goto 7
4 Let $r = r \dot{-} m$
5 Let $q = q + 1$
6 go to 3
7 continue

QED

Example 1.2 *The functions $Z = X + Y$, $Z = XY$, $Z = X^Y$, and $X \dot{-} Y$ are UR-Basic computable.*

Proof
$Z = X + Y$:

```
Let Z = X
For i = 1 to Y
    Let Z = Z + 1
Next i
```

$Z = XY$:

```
Let Z = 0
For i = 1 to Y
    Let Z = Z + X
Next i
```

$Z = X^Y$:

```
Let Z = 1
For i = 1 to Y
    Let Z = ZX
Next i
```

$Z = X \dot{-} Y$:

```
Let Z = X
For i = 1 to Y
    Let Z = Z -̇ 1
Next i
```

QED

Exercise 1.3. Prove that the greatest common divisor function $d = gcd(n, m)$ is UR-Basic computable. Or if you prefer the function $f(n) =$ the n^{th} prime. Or you can prove that your favorite function is UR-Basic computable.

2 Primitive recursive functions

The class of primitive recursive functions is the smallest set of functions $f : \omega^m \to \omega$ of arbitrary arity m which contain

1. the constant zero function, $Z : \omega \to \omega$, $Z(n) = 0$ all n,

2. the successor function, $S : \omega \to \omega$ with $S(n) = n+1$ all n (which we usually write $n+1$), and

3. the projections $\pi^n_m(x_1, \ldots, x_n) = x_m$ for $1 \leq m \leq n < \omega$

and is closed under

- composition: h is primitive recursive, if

  $$h(x_1, \ldots, x_m) = f(g_1(x_1, \ldots, x_m), \ldots, g_n(x_1, \ldots, x_m))$$

  where f is n-ary and each g_i is m-ary are primitive recursive, and

- primitive recursion: h is primitive recursive, if

  $$\begin{aligned} h(0, x_1, \ldots, x_m) &= g(x_1, \ldots, x_m) \\ h(y+1, x_1, \ldots, x_m) &= f(y, x_1, \ldots, x_m, h(y, x_1, \ldots, x_m)) \end{aligned}$$

  where g is m-ary and f is $(m+2)$-ary primitive recursive.

Note that by using the projections and compositions we may swap variables around and introduce dummy variables, e.g.

$$h(x, y, z) = f(g(x, y), z, k(z, x)) = f(g_1(x, y, z), g_2(x, y, z), g_3(x, y, z))$$

where

$$\begin{aligned} g_1(x, y, z) &= g(\pi^3_1(x, y, z), \pi^3_2(x, y, z)) \\ g_2(x, y, z) &= \pi^3_3(x, y, z) \\ g_3(x, y, z) &= k(\pi^3_3(x, y, z), \pi^3_2(x, y, z)) \end{aligned}$$

A predicate $P \subseteq \omega^n$ is primitive recursive iff its characteristic function $\chi_P(\vec{x})$ is where

$$\chi_P(\vec{x}) = \begin{cases} 1 & \text{if } P(\vec{x}) \\ 0 & \text{if } \neg P(\vec{x}) \end{cases}$$

Constant functions of any arity are primitive recursive. E.g., the function $f(x, y, z) = 2$ for all x, y, z is defined by

$$f(x, y, z) = S(S(Z(\pi^3_1(x, y, z))))$$

Define $z = x + y$:

$x + 0 = x$
$x + (y + 1) = (x + y) + 1$

Define $z = xy$:
$x0 = 0$
$x(y + 1) = xy + x$

Define $z = x^y$:
$x^0 = 1$
$x^{y+1} = x^y x$

Define $z = x^{(y)} = x^{x^{x^{\cdot^{\cdot^{x}}}}}$:
$x^{(0)} = x$
$x^{(y+1)} = x^{x^{(y)}}$

Define $z = x!$:
$0! = 1$
$(x + 1)! = (x + 1)x!$

Define $z = x \dot{-} 1$:
$0 \dot{-} 1 = 0$
$(x + 1) \dot{-} 1 = x$

Define $z = y \dot{-} x$:
$y \dot{-} 0 = y$
$y \dot{-} (x + 1) = (y \dot{-} x) \dot{-} 1$

Define
$$sign(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \end{cases}$$
by $sign(x) = 1 \dot{-} (1 \dot{-} x)$.

Proposition 2.1 *The predicates $x \leq y$, $x = y$, $x < y$ are primitive recursive. If P and Q are primitive recursive predicates, then so is $P \vee Q$ and $\neg P$. If $P(\vec{x}, y)$ is a primitive recursive predicate and $f(\vec{x})$ a primitive recursive function, then $Q(\vec{x}) \equiv P(\vec{x}, f(\vec{x}))$ is a primitive recursive predicate.*

Proof

$\chi_{\leq}(x,y) = 1 \dot{-} (x \dot{-} y)$

$\chi_{P\vee Q} = sign(\chi_P + \chi_Q)$

$\chi_{\neg P} = 1 \dot{-} \chi_P$

$x = y$ iff $x \leq y$ and $y \leq x$

$x < y$ iff $\neg y \leq x$

$\chi_Q(\vec{x}) = \chi_P(\vec{x}, f(\vec{x}))$

QED

Proposition 2.2 *If $P(\vec{x}, y)$ is a primitive recursive predicate and $f(\vec{x})$ a primitive recursive function, then*

$$\exists y \leq f(\vec{x})\ P(\vec{x}, y) \text{ and } \forall y \leq f(\vec{x})\ P(\vec{x}, y)$$

are both primitive recursive predicates.

Proof

Let

$$Q(\vec{x}, z) \equiv \exists y \leq z \ \ P(\vec{x}, y)$$

Then χ_Q has the recursive definition:

$\chi_Q(\vec{x}, 0) = \chi_P(\vec{x}, 0)$

$\chi_Q(\vec{x}, z+1) = sign(\chi_Q(\vec{x}, z) + \chi_P(\vec{x}, z+1))$

Note that

$$Q(\vec{x}, h(\vec{x}) \equiv \exists y \leq h(\vec{x})\ P(\vec{x}, y)$$

and

$$\forall y \leq h(\vec{x})\ P(\vec{x}, y) \equiv \neg\exists y \leq h(\vec{x})\ \neg P(\vec{x}, y)$$

QED

For example,

x divides y iff $\exists z \leq y\ y = xz$.

x is a Prime iff $x > 1$ and $\forall y \leq x$ if y divides x, then $y = 1$ or $y = x$.

are primitive recursive predicates.

Bounded search: define $f(\vec{x}, z) = \mu y \leq z\ P(\vec{x}, y)$ where f is the least $y \leq z$ which satisfies $P(\vec{x}, y)$ and $f = 0$ if no $y \leq z$ can be found.

Proposition 2.3 *Suppose Q is a primitive recursive predicate and h a primitive recursive function. Then*

$$g(\vec{x}) = \mu y \leq h(\vec{x})\ P(\vec{x}, y)$$

is primitive recursive.

Proof
Let

$$Q(\vec{x}, y) \equiv P(\vec{x}, y) \wedge \forall u < y\ \neg P(\vec{x}, u).$$

Then if we define

$$f(\vec{x}, z) = \mu y \leq z\ P(\vec{x}, y)$$

then

$$f(\vec{x}, z) = \sum_{y=0}^{z} y \cdot \chi_Q(\vec{x}, y)$$

which has the following primitive recursive definition:
$f(\vec{x}, 0) = 0$
$f(\vec{x}, z+1) = f(\vec{x}, z) + (z+1)\chi_Q(\vec{x}, z+1)$
Hence

$$g(\vec{x}) = f(\vec{x}, h(\vec{x})) = \mu y \leq h(\vec{x})\ \ P(\vec{x}, y).$$

QED

Proposition 2.4 *If $f : \omega \to \omega$ is primitive recursive, the graph(f) is a primitive recursive predicate. If graph(f) is a primitive recursive predicate and there is a primitive recursive function g which bounds f, then f is primitive recursive.*

Proof
Graph(f) has characteristic function $\chi_=(f(\vec{x}), y)$. If f is bounded by g then

$$f(\vec{x}) = \mu y \leq g(\vec{x})\ \ (\vec{x}, y) \text{ is in the graph of } f.$$

QED
Examples:
z=max(x,y) iff ($x = z$ and $x \geq y$) or ($y = z$ and $y \geq x$)
has primitive recursive graph and is bounded by $x + y$, so it is a primitive recursive function.

Division,Quotient: input $n, m > 0$ output q, r with $n = qm+r$ and $r < m$. $q = \text{quotient}(n, m)$ and $r = \text{remainder}(n, m)$ both have primitive recursive graphs bounded by $n + m$ so they are primitive recursive.

Exercise 2.5. Let $r(n) = n^{th}$ digit of $\sqrt{2} = 1.4142136\ldots$, so $r(0) = 1$, $r(1) = 4$, and so on. Prove that r is primitive recursive. If you prefer you may use $e = 2.7182818\ldots$ instead of $\sqrt{2}$. Does every naturally occurring constant in analysis have this property?

Exercise 2.6. Define n is square-free iff $n \geq 2$ and no m^2 divides n for $m \geq 2$. Let $S(n)$ be the sum of the first n square-free numbers. Prove S is a Primitive recursive function.

3 Primitive recursive functions are UR-Basic computable

Theorem 3.1 *Every primitive recursive function is UR-Basic computable.*

Proof
The empty program with input x and output y, computes the constant zero function. Similarly for the projections. The successor function is computed by the one-line program "Let x=x+1", with input and output variable x.

For closure under composition: $z = f(g_1(\vec{x}), \ldots, g_n(\vec{x}))$ use the basic program:

Let $z_1 = g_1(\vec{x})$
Let $z_2 = g_2(\vec{x})$
. . .
Let $z_n = g_n(\vec{x})$
Let $y = f(z_1, \ldots, z_n)$

where appropriate substitution of UR-Basic code has been done.

The basic code for a primitive recursive definition

$f(\vec{x}, 0) = g(\vec{x})$
$f(\vec{x}, n + 1) = h(n, f(\vec{x}, n), \vec{x})$

looks like

input $\vec{x}, n$
Let $y = g(\vec{x})$

For $i = 1$ to n
    Let $y = h(i \dot{-} 1, y, \vec{x})$
next i
output $y = f(\vec{x}, n)$
QED

4 UR-BASIC computable functions are recursive

Definition 4.1 *The partial recursive functions are the smallest class of functions containing the primitive recursive functions and closed under composition, primitive recursion, and unbounded search μ:*

$$f(\vec{x}) = \mu y \;\; P(\vec{x}, y)$$

where P is a recursive predicate,i.e., its characteristic function is recursive.

Theorem 4.2 *(Kleene) There exists a primitive recursive predicate Q and primitive recursive g such that for every partial UR-Basic computable $f : \omega \to \omega$ there exists an e such that*

$$\forall x \quad f(x) = g(\mu z \;\; Q(e, x, z)).$$

Proof
An informal description of g and Q are as follows. $Q(e, x, z)$ says that the program coded by e with input x does the computation coded by z. $g(z)$ is the value of the output variable at the final step of the computation coded by z.

In order to more formally define Q we begin by describing a method of coding pairs and finite sequences using primitive recursive functions.
Coding pairs. the mapping $x, y \mapsto \langle x, y \rangle$ defined by

$$\langle x, y \rangle = 2^x(2y + 1) - 1$$

is a primitive recursive bijection between ω^2 and ω. Both unpairing functions are primitive recursive since if $x = \langle x_0, x_1 \rangle$, then $x_0, x_1 \leq x$. So define the head and tail functions h and t as follows:

$$h(\langle x, y \rangle) = x \text{ and } t(\langle x, y \rangle) = y$$

Triples can be coded by $\langle x, y, z\rangle = \langle x, \langle y, z\rangle\rangle$ and similarly by induction for n-tuples:

$$\langle x_1, x_2, \ldots, x_n\rangle = \langle x_1, \langle x_2, \ldots, x_n\rangle\rangle.$$

Note that, for example,

$$h(t(t(\langle x, y, z, w\rangle))) = z$$

so the "coordinate function" $\langle x, y, z, w\rangle \mapsto z$ is primitive recursive.
To code finite sequences of arbitrary length define the function

$$c(y, k) = h(t^{(k)}(y))$$

where $t^{(k)}$ stands for the composition of t with itself k times. It has a primitive recursive definition $f(k, x) = t^{(k)}(x)$:

$$f(x, 0) = x$$

$$f(x, k + 1) = t(f(x, k))$$

It is easy to check that c has the property that for any n and for any finite sequence $y_0, y_1, \ldots, y_n$ there exists y such that $c(y, k) = y_k$ for all $k \leq n$. We often use y_i to denote $c(y, i)$

We can assume that the UR-Basic program only uses the variable v_i for $i < \omega$ and that the input variable is v_0 and output variable v_1.

1. $S = \langle 0, i\rangle \in \omega$ codes the statement "Let $v_i = v_i + 1$".
2. $S = \langle 1, i\rangle \in \omega$ codes the statement "Let $v_i = v_i \dot{-} 1$".
3. $S = \langle n, i, j, k\rangle$ for $n \geq 2$ codes the statement "If $v_i \leq v_j$ then goto k".

For $e \in \omega$ let $e = \langle n, S\rangle$ and let $S_0, S_1, \ldots, S_{n-1}$ be the program statements with S_i coded by $c(S, i)$.

Next we define three primitive recursive predicates:

In the tuple (e, x, y), e codes the program, x is the input value and y is pair $\langle k, V\rangle$ coding the line k in the program which is being executed and V coding the values of the variables.

$$Init(e, x, y) \equiv$$

$$\exists V < y \;\; y = \langle 0, V\rangle \text{ and } c(V, 0) = x \text{ and } \forall i < e \; (i > 0 \rightarrow c(V, i) = 0)$$

Since this is the start we want to start with Statement 0, i.e., $y = (0, V)$ and $v_0 = x$ and $v_i = 0$ for all i with $0 < i < e$. Note that we can bound this by e since e cannot refer to any variables with index higher than e.

$$Halt(e, y) \equiv$$
$$\exists n, S < e\ \exists k, V < y\ \ y = \langle k, V\rangle \text{ and } e = \langle n, S\rangle \text{ and } k \geq n$$

All this says is we halt when we try to execute a line number greater than the length of the program.

$$Onestep(e, y, y') \equiv$$

(This says we take one step from y to y'.)

$\exists k, V, k', V' < y + y'$ and $\exists n, S < e$ such that all of the following are true:

1. $y = \langle k, V\rangle$, $y' = \langle k', V'\rangle$, and $e = \langle n, S\rangle$
2. $k < n$ (we don't take a step if program has halted)
3. If $c(S, k)$ codes "Let $v_i = v_i + 1$" then
   $c(V', i) = c(V, i) + 1$,
   $c(V', j) = c(V, j)$ for all $j < e$ with $j \neq i$, and
   $k' = k + 1$,
4. If $c(S, k)$ codes "Let $v_i = v_i \dot{-} 1$" then
   $c(V', i) = c(V, i) \dot{-} 1$,
   $c(V', j) = c(V, j)$ for all $j < e$ with $j \neq i$, and
   $k' = k + 1$.
5. If $c(S, k)$ codes "If $v_i \leq v_j$ then goto l" then
   $V = V'$ and
   if $c(V, i) \leq c(V, j)$ then $k' = l$ else $k' = k + 1$.

Next we define the predicate $Q(e, x, z)$. Informally, it says that z codes a computation using program e and input x.

$$Q(e, x, z) \equiv$$
$$\exists N, y < z\ \ z = \langle N, y\rangle \text{ and } Init(e, x, c(y, 0)) \text{ and } Halt(e, c(y, N)) \text{ and}$$
$$\forall i < N\ \ Onestep(e, c(y, i), c(y, i + 1))$$

Finally we define the function g. It simply extracts the value of v_1 the output variable from the computation coded by z. Since $g(z) \leq z$ it is enough to see that its graph is primitive recursive:

$$g(z) = v \text{ iff}$$
$$\exists N, y, V, k < z \;\; \langle N, y\rangle = z \text{ and } c(y, N) = \langle k, V\rangle \text{ and } c(V, 1) = v$$

QED

Corollary 4.3 *The family of (partial) UR-Basic computable functions is the same as the family of (partial) recursive functions.*

Proof
The family of UR-Basic computable functions is closed under unbounded search μ, i.e.,

To compute the function $f(\vec{x}) = \mu y \;\; P(\vec{x}, y)$ use code:

1 Let $y = 0$
2 If $P(\vec{x}, y)$ then goto 5
3 Let $y = y + 1$
4 Goto 2
5 continue

Hence every partial recursive function is partial UR-Basic computable. The Theorem supplies the other inclusion.
QED

The Theorem shows that only one unbounded search is needed to get every partial recursive function. Something that is not immediately evident from the definition of recursive function.

Exercise 4.4. Another way to code finite sequences of arbitrary length is to use prime factorization.

(a) Define: nextprime(x) $= y$ to be the smallest prime $y > x$. Prove that nextprime(x) is primitive recursive.

(b) Define: $p_0 = 2$ and p_n is the n^{th} odd prime. Prove that the function $n \mapsto p_n$ is primitive recursive.

(c) Define $c(x, i) = k$ iff k is the least integer such that p_i^{k+1} does not divide x. Prove that c is primitive recursive and for any finite sequence $x_0, x_1, \ldots, x_n$ there exists x such that $c(x, k) = x_k$ for all $k \leq n$.

Exercise 4.5. Suppose that $f : \omega \to \omega$ is UR-Basic computable by a program P and there exists a primitive recursive function $s : \omega \to \omega$ such that for every x the program P computes $f(x)$ in $\leq s(x)$ steps. Prove that f is primitive recursive.

Exercise 4.6. The programming language P-Basic has only four kinds of statements

(a) Let $X = X + 1$
(b) Let $X = X \dot{-} 1$
(c) Let $X = Y$

where X, Y are any variables

(d) for-next loops, e.g.

For $i = 1$ to n
S_1
$\vdots$
S_k
Next i

The loop variable i and n must be distinct and in the body of the loop $(S_1, \ldots, S_k)$ the variables i and n are not allowed to be changed, i.e.,

For n= 1 to ...
For i= 1 to ...
Let n = ...
Let i = ...

are not allowed. Prove that the P-Basic computable functions are the same as the primitive recursive functions.

Exercise 4.7 *Another popular pairing function* $p : \omega^2 \to \omega$ *is described by Figure 1. Show that* p *is a polynomial. Hint: the point* (m, n) *is on the diagonal of the square of area* $(m + n)^2$.

5 Church-Turing Thesis

Church-Turing Thesis:

Every intuitively computable function is recursive.

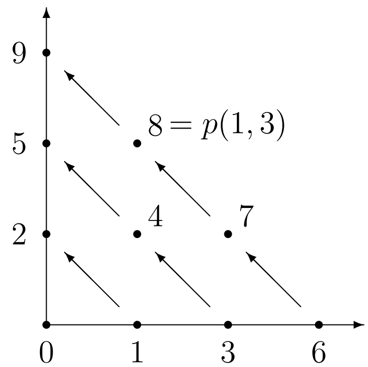


Figure 1: Pairing function $p(n, m)$, see exercise 4.7.

Good evidence for Church's thesis is the fact that all other ways people have come up with to formalize the notion of effectively computable function (e.g. RAM machines, register machines, generalized recursive functions, neural nets, etc) can be shown to define the same set of functions. Church's original formal definition was using the lambda calculus. However it is not easy to see that even the elementary arithmetic functions such as successor or addition are representable in the lambda calculus. It took his student, Kleene, several weeks to prove this. Similarly, it is also true that all computable functions can be represented in John Conway's Game of Life. But this is difficult to see and so does not really give convincing evidence that the informal notion of effectively calculable has been captured.

In section 45 we define the notion of Turing computable function and include Turing's analysis of why every effectively calculable function should be Turing computable.

Proposition 5.1 *There exists a recursive function $f : \omega \to \omega$ which is not primitive recursive.*

Proof
Make an effective list $f_n : \omega^{k_n} \to \omega$ of all the primitive recursive functions. Define $f(n) = f_n(n) + 1$ if f_n is a 1-ary function, otherwise put $f(n) = 0$. Since the listing is effective by the Church-Turing Thesis the function f is recursive. But by the usual diagonal argument f is not on the list.

QED

Exercise 5.2. Prove that there exists a (total) $h : \omega \to \omega$ whose graph is a primitive recursive predicate but h is not a primitive recursive function. Hint: consider $h(x) = \mu z\, Q(e, x, z)$.

Exercise 5.3. Prove there exists a primitive recursive bijection $p : \omega \to \omega$ such that p^{-1} is not primitive recursive.

6 Universal partial recursive function

Proposition 6.1 *(Turing) There exists a universal partial recursive function*

$$\psi : \omega \to \omega$$

i.e. if we define $\psi_e(x) = \psi(\langle e, x\rangle)$ *then* $\{\psi_e : e \in \omega\}$ *is a uniformly computable listing of all partial recursive functions.*

Proof
$\psi(\langle e, x\rangle) = g(\mu z\, Q(e, x, z))$.
QED

Note that for any $n \geq 2$ if $f(x_1, \ldots, x_n)$ is a partial recursive function then there will be e such that

$$\forall x_1, \ldots, x_n \;\; \psi(\langle e, \langle x_1, \ldots, x_n\rangle\rangle) = f(x_1, \ldots, x_n).$$

So ψ is universal for partial recursive functions of any arity.

Proposition 6.2 *(Padding Lemma) There exists a 1-1 recursive function p such that* $\psi_e = \psi_{p(e,n)}$ *for every* e, n.

Proof
To pad the program $S_0, S_1, \ldots, S_m$ coded by e just add the statement

$$S_{m+1} = \text{LetDonothing}\langle \text{e}, \text{n}\rangle = \text{Donothing}\langle \text{e}, \text{n}\rangle + 1$$

and let $p(e, n)$ code this new program.
QED

Proposition 6.3 *(S-n-m Theorem). There exists a recursive function S such that* $\psi_e(\langle x, y\rangle) = \psi_{S(e,x)}(y)$ *for all* e, x, y.

Proof
Given $\mathcal{P}$ the program coded by e and input x make-up a new program coded by $S(e,x)$ which puts x into $\mathcal{P}$'s first input variable and then pops into program $\mathcal{P}$.
QED

The name S-n-m comes from the obvious generalization to n-tuple $\vec{x}$ and m-tuple $\vec{y}$

$$\psi_e(\langle \vec{x}, \vec{y} \rangle) = \psi_{S_{n,m}(e,\vec{x})}(\vec{y})$$

so what we are stating is the S-1-1 Theorem.

These propositions can be combined as follows:

Proposition 6.4 *Suppose $\theta(x,y)$ is a partial recursive function. Then there is a one-to-one recursive function $f : \omega \to \omega$ such that*

$$\forall x, y \quad \psi_{f(x)}(y) = \theta(x,y).$$

Proof
Suppose $\theta = \psi_{e_0}$. Then

$$\theta(x,y) = \psi_{p(S(e_0,x),x)}(y)$$

and so $f(x) = p(S(e_0,x),x)$ works.
QED

We call this the 1-1-S-1-1 Theorem.

7 The recursively enumerable sets

Definition 7.1 *For $A \subseteq \omega$ define:*

1. *A is recursively enumerable iff either A is empty or A is the range of a recursive function, i.e., $A = \{a_0, a_1, a_2, \ldots\}$ where the function $n \mapsto a_n$ is recursive. This is abbreviated r.e.*
2. *A is Σ^0_1 iff there exists a recursive predicate $R \subseteq \omega^2$ such that*

$$A = \{n \ : \ \exists m \ R(n,m)\}.$$

Definition 7.2 $W = \{\langle e,x \rangle \ : \ \psi(\langle e,x \rangle) \downarrow\}$. *Then* $\{W_e \ : \ e \in \omega\}$ *where* $W_e = \{x \ : \ \langle e,x \rangle \in W\}$ *is a uniform listing of the r.e. sets.*

Proposition 7.3 *For $A \subseteq \omega$ the following are equivalent:*
(1) A is recursively enumerable.
(2) A is the domain of a partial recursive function.
(3) A is Σ^0_1.
(4) A is finite or A has a one-to-one recursive enumeration.
(5) There exists e such that $A = W_e$.

Proof

(1) $\rightarrow$ (2): Given a recursive enumerable listing a_n describe a partial recursive function f by:

- input x
- look for x on the list: $a_0, a_1, a_2, \ldots$
- halt if you find it, otherwise continue looking forever.

(2) $\rightarrow$ (1): Define $\psi_{e,s}(x)\downarrow= y$ to mean that

$$e, x, y < s \quad \wedge \quad \exists z < s \; (Q(e,x,z) \quad \wedge \quad g(z) = y).$$

See Theorem 4.2. The predicate

$$P(e,x,y,s) \equiv \psi_{e,s}(x)\downarrow= y$$

is primitive recursive. It roughly says that the algorithm coded by e with input x terminates in fewer than s steps and outputs y. (Actually z is a sequence coding the values of the variables and the line number at each step.) If A is the domain of ψ_e, then either A is empty or let $x_0 \in A$ be arbitrary and define a recursive enumeration of A by

$$a_n = \begin{cases} x & \text{if } n = \langle x, y, s\rangle \text{ and } \psi_{e,s}(x)\downarrow= y \\ x_0 & \text{otherwise.} \end{cases}$$

(1) $\rightarrow$ (3): Let $f : \omega \rightarrow \omega$ be recursive and have range A. Let R be the graph of f, then $y \in A$ iff $\exists x \; R(x,y)$.

(3) $\rightarrow$ (2): Suppose $x \in A$ iff $\exists y \; R(x,y)$. Then $f(x) = \mu y \; R(x,y)$ is partial recursive with domain A.

(1) $\rightarrow$ (4): Given $\{a_n \; : \; n < \omega\}$ a *recursive* enumeration of A, define a recursive enumeration $\{b_n \; : \; n < \omega\}$ by:

$b_{n+1} = a_m$ where m is the least such that $a_m \notin \{b_i : i \leq n\}$.

(2) $\leftrightarrow$ (5): by definition.

QED

Definition 7.4 *For $A \subseteq \omega$, define:*

1. *A is recursive iff its characteristic function χ_A is recursive.*
2. *$\overline{A} = \omega \setminus A$ the complement of A,*
3. *A is Π^0_1 iff $\overline{A}$ is Σ^0_1, and*
4. *$\Delta^0_1 = \Sigma^0_1 \cap \Pi^0_1$.*

Proposition 7.5 *For $A \subseteq \omega$ the following are equivalent:*
(1) A is recursive.
(2) A and $\overline{A}$ are both recursively enumerable.
(3) A is Δ^0_1.
(4) A is finite or A has a strictly increasing recursive enumeration.

Proof
(1) $\rightarrow$ (2): It is easy to see that recursive implies recursively enumerable and that the complement of a recursive set is recursive.
(2) $\rightarrow$ (1): Input x. Effectively list A and $\overline{A}$ simultaneously until x shows up.
(2) iff (3): Trivial.
(1) $\rightarrow$ (4): Take a_n to be the n^{th} element of A.
(4) $\rightarrow$ (1): Let $\{a_n : n < \omega\}$ be a strictly increasing *recursive* enumeration of A. The following algorithm computes the characteristic function of A:

- Input x.
- Find n such that $a_n > x$.
- Then $x \in A$ iff $x \in \{a_i : i < n\}$.

QED

Example 7.6 *There exists an r.e. set K which is not recursive.*

Proof
$$K = \{e : \psi_e(e)\downarrow\}$$
If $\overline{K}$ is the domain of ψ_e, then $e \in K$ iff $e \notin K$.
QED

Proposition 7.7 *Every infinite r.e. set contains an infinite recursive set.*

Proof
Given $\{a_n : n < \omega\}$ a *recursive* enumeration of A, define a strictly increasing recursive enumeration $\{b_n : n < \omega\}$ by:
$b_0 = a_0$ and
$b_{n+1} = a_m$ where m is the least such that $a_m > b_n$.
QED

Proposition 7.8 *If A and B are r.e. sets, then $A \cap B$ is r.e. and $A \cup B$ is r.e. If A and B are recursive sets, then $A \cap B$, $A \cup B$, and $\overline{A}$ are all recursive sets.*

Proof
Domain of $f + g$ is the intersection of domain f and domain g. Enumerate $A \cup B$ by $x_{2n} = a_n$ and $x_{2n+1} = b_n$.
QED

Exercise 7.9. Suppose that $V \subseteq \omega$ is r.e. For each n define $V_n = \{x : \langle n, x\rangle \in V\}$. Prove that $\cup_n V_n$ is r.e.

Exercise 7.10. Prove that every nonempty recursively enumerable set A is the range of a primitive recursive function. Extra Credit: prove that not every infinite recursively enumerable set is the range of a one-to-one primitive recursive function.

Exercise 7.11. (a) For a partial function $f : \omega \to \omega$ prove that f is partial recursive iff its graph is recursively enumerable.

(b) For a partial recursive h prove there is a partial recursive g with $dom(g) \supseteq range(h)$ such that

$$\forall y \in range(h) \quad h(g(y)) = y.$$

(c) Give an example for (b) for which g cannot be total.

Exercise 7.12. Consider a partial function $f : \omega \to \omega$ and the three set:

1. $dom(f) \subseteq \omega$
2. $graph(f) \subseteq \omega \times \omega$

3. $range(f) \subseteq \omega$.

For each of the sets $(1), (2), (3)$ could be:
(a) recursive or
(b) recursively enumerable but not recursive.
For each of the 8 possibilities, either give an example of such an f or prove there is no such f. Extra credit: consider the third possibility (c) not recursively enumerable.

Exercise 7.13. If $f : \omega \to \omega$, then f^n denotes f applied n times; e.g., $f^3(0) = f(f(f(0)))$. Give an example of a (total) recursive f such that $\{f^n(0) : n \in \omega\}$ is not recursive.

Exercise 7.14. Define $V_e = \{x : \langle e, x\rangle \in V\}$. Prove or disprove:

1. $\exists V$ recursively enumerable such that $\{V_e : e \in \omega\}$ is the set of all recursive sets.
2. $\exists V$ recursive such that $\{V_e : e \in \omega\}$ is the set of all recursive sets.
3. $\exists V$ r.e. such $\{V_e : e \in \omega\}$ is the set of all nonempty r.e. sets.
4. $\exists f$ a recursive function such that for all $e \quad W_e \neq \emptyset$ implies $f(e) \in W_e$.
5. $\exists f$ partial recursive such that for all $e \quad W_e \neq \emptyset$ implies $f(e) \downarrow \in W_e$.

Exercise 7.15 *Prove there exists a recursive function $f : \omega \to \omega$ such that for every e*

$$W_e \textit{ infinite } \to (\psi_{f(e)} : \omega \to W_e \textit{ is total, one-to-one, and onto}).$$

For the definition of W_e see Definition 7.2.

8 Separation and reduction

Example 8.1 *There exists disjoint r.e. sets K_0 and K_1 which are recursively inseparable, i.e., there is not exists a recursive set $R \subseteq \omega$ with $K_0 \subseteq R$ and $K_1 \subseteq \overline{R}$.*

Proof

$$K_0 = \{e : \psi_e(e) \downarrow= 0\} \text{ and } K_1 = \{e : \psi_e(e) \downarrow= 1\}$$

QED

Definition 8.2 *For any $\Gamma \subseteq P(\omega)$ define $\widetilde{\Gamma}$ to be the set of all $\overline{A}$ for $A \in \Gamma$ and define $\Delta = \Gamma \cap \widetilde{\Gamma}$. $Sep(\Gamma)$ is the property that for every $A, B \in \Gamma$ disjoint there exists $C \in \Delta$ with $A \subseteq C$ and $B \subseteq \overline{C}$. $Red(\Gamma)$ (the reduction principle) is the property that for every $A, B \in \Gamma$ there exists disjoint $A' \subseteq A$ and $B' \subseteq B$ with $A', B' \in \Gamma$ and $A \cup B = A' \cup B'$.*

Proposition 8.3 *$Red(\Gamma)$ implies $Sep(\widetilde{\Gamma})$.*

Proof
Apply reduction to the complements.
QED

Proposition 8.4 *$Red(\Sigma^0_1)$ and hence $Sep(\Pi^0_1)$.*

Proof
$A = \{x : \exists u\ R(u, x)\}$ and $B = \{x : \exists v\ S(v, x)\}$. Put

$$x \in A' \leftrightarrow \exists u\ R(u, x) \text{ and } \forall v \leq u \neg S(v, x)$$

$$x \in B' \leftrightarrow \exists v\ S(v, x) \text{ and } \forall u < v \neg R(u, x)$$

QED

In example 8.1 it follows that K_0 and K_1 cannot be separated by disjoint Π^0_1 sets B_0 and B_1 because such a B_0 and B_1 could be recursively separated.

Exercise 8.5. Prove $Sep(\Gamma)$ for $\Gamma = \{A \cup B : A \in \Sigma^0_1,\ B \in \Pi^0_1\}$.

9 Many-one reducibility

Definition 9.1 *For $A, B \subseteq \omega$ define:*

1. *$A \leq_m B$ iff there exists a recursive function f such that*

$$\forall x \in \omega\ \ x \in A \leftrightarrow f(x) \in B.$$

Equivalently, $f^{-1}(B) = A$. Also equivalently $f(A) \subseteq B$ and $f(\overline{A}) \subseteq \overline{B}$.

2. *$A \le_1 B$ iff the f in the definition of $\le_m$ can be taken to be one-to-one.*

Proposition 9.2 1. *$A \le_1 B$ implies $A \le_m B$.*

2. *$A \le_m B$ iff $\overline{A} \le_m \overline{B}$ and similarly for $\le_1$.*
3. *$\le_m$ and $\le_1$ are transitive and reflexive.*
4. *$A \le_m B$ and B is recursive, then A is recursive.*
5. *$A \le_m B$ and B is recursively enumerable, then A is recursively enumerable.*

Proof
Most of these are trivial. Note that f reduces A to B then it also reduces $\overline{A}$ to $\overline{B}$. Transitivity follows by composition.

For (4) if f witnesses $A \le_m B$, i.e.,

$$\forall n \;\; n \in A \text{ iff } f(n) \in B,$$

then $\chi_A(n) = \chi_B(f(n))$.

For (5) suppose that

$$n \in B \text{ iff } \exists m \;\; R(n, m)$$

and

$$\forall n \;\; n \in A \text{ iff } f(n) \in B.$$

Then

$$\forall n \;\; n \in A \text{ iff } \exists m \;\; R(f(n), m).$$

QED

Definition 9.3 1. *$A \equiv_m B$ iff $A \le_m B$ and $B \le_m A$.*

2. *$m - deg(A) = \{B \; : \; A \equiv_m B\}$, the many-one degree of A.*
3. *$A \equiv_1 B$ iff $A \le_1 B$ and $B \le_1 A$.*
4. *$1 - deg(A) = \{B \; : \; A \equiv_1 B\}$, the one degree of A.*

Exercise 9.4 *Suppose A and B are infinite r.e. sets and $A \le_1 B$. Show there is a computable one-to-one reduction of A to B which maps A onto B.*

10 Rice's index Theorem

Recall that $\{W_e : e \in \omega\}$ is the standard listing of all r.e. sets (7.2).

Example 10.1 *Empty* $= \{e : W_e = \emptyset\}$ *is not recursive.*

Proof
Define

$$\theta(e,x) = \begin{cases} \downarrow = 0 & \text{if } e \in K \\ \uparrow & \text{otherwise} \end{cases}$$

By the S-n-m theorem there exists f recursive such that

$$\forall e, x \quad \psi_{f(e)}(x) = \theta(e,x)$$

But then $e \in K$ iff $W_{f(e)} \neq \emptyset$ iff $f(e) \notin E$ so $K \leq_m \overline{E}$ and therefore E not recursive.
QED

Proposition 10.2 *(Rice) If A is a nontrivial index set, then A is not recursive.*

Proof
This is like the proof for Empty. Without loss of generality assume the index of the empty function is in A and the index e_0 of some nonempty partial computable function is not in A. Define

$$\theta(e,x) = \begin{cases} \psi_{e_0}(x) & \text{if } e \in K \\ \uparrow & \text{otherwise} \end{cases}$$

By the S-n-m theorem there exists f recursive such that

$$\forall e, x \quad \psi_{f(e)}(x) = \theta(e,x)$$

But then

$$e \in K \text{ iff } f(e) \notin A$$

and therefore A is not recursive.
QED

11 Myhill's recursive permutation Theorem

Theorem 11.1 *(Myhill) $A \leq_1 B$ and $B \leq_1 A$ iff there exists a recursive bijection $\pi : \omega \to \omega$ with $\pi(A) = B$.*

Proof
The Schroeder-Bernstein Theorem says: if there exists a 1-1 $f : A \to B$ and 1-1 $g : B \to A$, then there exists a bijection $h : A \to B$. One way to prove this is to assume A and B are disjoint and define a bipartite graph on the vertices $A \cup B$. Put $a \in A$ connected to b iff either $f(a) = b$ or $g(b) = a$. As f and g are 1-1 the order of every vertex is either 1 or 2. The connected components of this graph come in 4 types, see figure 2. Note that in Type 1 the point $a \in A$ is not in the range of g and in Type 2 the point $b \in B$ is not in the range of f. Type 4 components are infinite in both 'directions' while Type 3 is the only finite component.

To get h simply define $h = f$ on any component of type 1,3, or 4 and $h = g^{-1}$ on components of type 2.

The proof of Myhill's theorem is similar except we may never know exactly which type of component we are looking at.

Suppose f and g are 1-1 recursive functions reducing A to B and B to A.

Effectively construct a sequence π_s of bijections with

1. $\pi_s : D_s \to E_s$ is a bijection.
2. D_s and E_s are finite subsets of ω.
3. $\pi_s \subseteq \pi_{s+1}$.
4. $n \in D_{2n}$ and $n \in E_{2n+1}$.
5. if $\pi_s(n) = m$, then either $m = fgfg \cdots fn$ or $n = gfgf \cdots gm$.

In the condition 5 we have dropped the parentheses to make it more readable.

If we then take $\pi = \cup_s \pi_s$, then π is a recursive bijection since we effectively constructed the sequence. It takes A to B, because suppose $\pi(n) = m$. Then if $m = fgfg \cdots fn$

$$n \in A \text{ iff } fn \in B \text{ iff } gfn \in A \text{ iff } fgfn \in B \text{ iff } \cdots \text{ iff } m = fgfg \cdots fn \in B$$

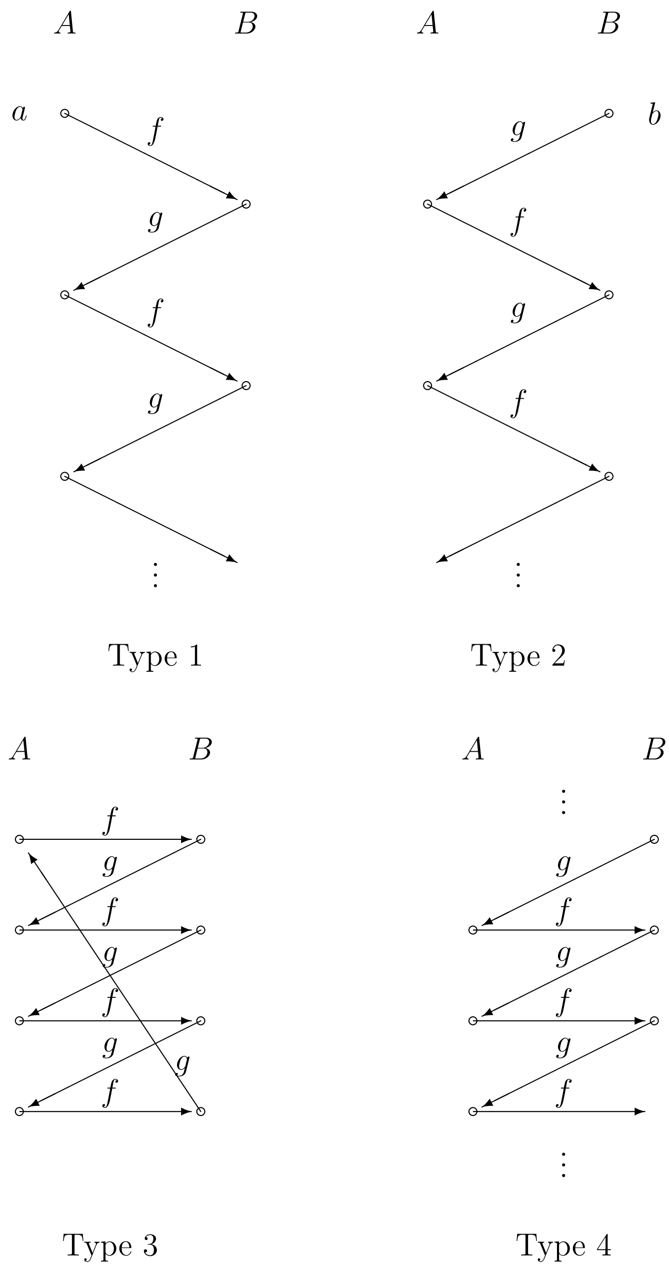


Figure 2: Schroeder-Bernstein connected components

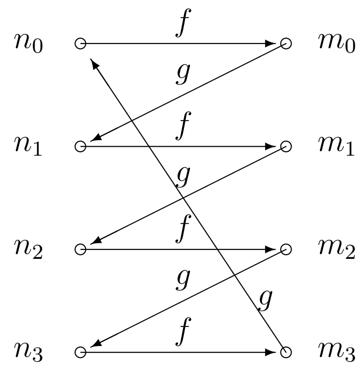


Figure 3: Myhill back and forth

similarly if $n = gfgf\cdots gm$

$m \in B$ iff $gm \in A$ iff $fgm \in B$ iff $gfgm \in A$ iff $\cdots$ iff $n = gfgf\cdots gm \in A$

either way $n \in A$ iff $m \in B$.

At stage s=0 we take π_0 to be the empty function.

At stage s+1 suppose we are given $\pi_s : D_s \to E_s$. If $s = 2n$ we try to extend π_s to include $n \in D_{s+1}$. If its already there we let $\pi_{s+1} = \pi_s$. Otherwise consider the following sequences:

Let $n = n_0$, $fn_0 = m_0$ and in general $f(n_k) = m_k$ and $g(m_k) = n_{k+1}$, see figure 3.

Case 1. For some k we have that $m_k \notin E_s$.

In this case we put $\pi_{s+1} = \pi_s \cup \{\langle n_0, m_k\rangle\}$.

Case 2. Not case 1.

In this case the connected component of the graph (see Figure 2) must be of Type 3, i.e., a finite closed loop. Suppose $g(m_k) = n_0$. But by condition 5 if all the m_k are in E_s, then they must map via π_s^{-1} to the set $\{n_0, n_1, \ldots, n_k\}$ (although not in any particular order). But this is a contradiction, since $n = n_0 \notin D_s$. Hence Case 2 cannot happen.

The construction at stage s+1 where $s = 2n + 1$ is entirely analogous except we make sure $n \in E_{s+1}$.

QED

Exercise 11.2. Define

$$Q = \{\langle e_1, e_2\rangle \ : \ e_1 \in W_{e_2},\ e_2 \in W_{e_1},\ \text{and } e_1 \neq e_2\}.$$

Prove that Q is creative.

12 Roger's adequate listing Theorem

Theorem 12.1 *(Rogers) Suppose $\rho : \omega \to \omega$ is partial recursive and we define $\rho_e(x) = \rho(e, x)$. Suppose*

1. *ρ is universal, i.e., $\{\rho_e \ : \ e \in \omega\}$ includes all partial recursive functions.*
2. *ρ satisfies padding, i.e., there exists one-to-one recursive $p : \omega \times \omega \to \omega$ such that*
$$\forall e, n \ \ \rho_e = \rho_{p(e,n)}$$
3. *ρ satisfies S-1-1, i.e., there exists a recursive $S : \omega \times \omega \to \omega$ such that*
$$\forall e_1, e_2, x \ \ \rho_{e_1}(\langle e_2, x\rangle) = \rho_{S(e_1,e_2)}(x)$$

Then there exists a recursive bijection $\pi : \omega \to \omega$ such that

$$\forall e \ \ \psi_e = \rho_{\pi(e)}$$

Proof
Let $\psi = \rho_{e_0}$. Using padding and S-1-1 for ρ we can find a 1-1 recursive function $f(e) = p(S(e_0, e))$ such that

$$\forall e \ \ \psi_e = \rho_{S(e_0,e)} = \rho_{f(e)}$$

similarly there is a 1-1 recursive function g such that

$$\forall e \ \ \rho_e = \psi_{g(e)}.$$

By the proof of Theorem 11.1 there is a recursive bijection $\pi : \omega \to \omega$ with the property that whenever $\pi(n) = m$ then either $m = fgfg\cdots fn$ or $n = gfgf\cdots gm$. But

$$\psi_n = \rho_{fn} = \psi_{gfn} = \ldots = \rho_{fgfg\cdots fn} = \rho_m$$

and

$$\rho_m = \psi_{gm} = \rho_{fgm} = \ldots = \psi_{gfgf\cdots gm} = \psi_n$$

so in either case $\psi_n = \rho_{\pi(n)}$.
QED

Exercise 12.2. Find an example of a partial recursive ρ which is universal but fails to satisfy padding. Find an example which is universal, satisfies padding but fails to satisfy S-1-1. (S-1-1 implies padding see Soare p.25-26.)

13 Kleene's Recursion Theorem

Theorem 13.1 *(Kleene - Recursion Theorem) For any recursive function f there exists an e with $\psi_e = \psi_{f(e)}$.*

Proof
Define a partial recursive function θ by

$$\theta(u, x) = \psi_{\psi_u(u)}(x) = \psi(\langle \psi(\langle u, u \rangle), x \rangle)$$

By padding-S-1-1 we can find a (one-to-one) recursive function $d : \omega \to \omega$ such that

$$\forall u \;\; \psi_{d(u)}(x) = \theta(u, x)$$

Let v be an index for $f \circ d$, i.e.,

$$\forall x \;\; \psi_v(x) = f(d(x))$$

Put $e = d(v)$ then

$$\psi_e(x) = \psi_{d(v)}(x) = \theta(v, x) = \psi_{\psi_v(v)}(x) = \psi_{f \circ d(v)}(x) = \psi_{f(e)}(x)$$

QED

From the proof we can get an infinite recursive set of fixed points e, since we can take any v' such that $\psi_{v'} = f \circ d$ and set $e' = d(v')$. Also note that our fixed point e is obtained effectively from an index for f, so given a recursive $f : \omega \times \omega \to \omega$ if we let $f_n : \omega \to \omega$ be defined by $f_n(x) = f(n, x)$ then we get a fixed points e_n

$$\psi_{e_n} = \psi_{f_n(e_n)}$$

and the function $h(n) = e_n$ is recursive. This is called the recursion theorem with parameters:

Theorem 13.2 *For any recursive function $f : \omega^2 \to \omega$ there exists a 1-1 recursive function $h : \omega \to \omega$ such that $\psi_{h(x)} = \psi_{f(x,h(x))}$ for all x.*

Example 13.3 *There are infinitely many e such that $\psi_e(0) = e$. There are infinitely many e such that $W_e = \{e\}$.*

Proof
Define $\theta(e, x) = e$ for all e. By the S-n-m Theorem there exists a recursive f such that

$$\forall e, x \ \ \psi_{f(e)} = \theta(e, x)$$

By the Recursion Theorem there are infinitely many fixed points for f, i.e.,

$$\psi_e = \psi_{f(e)}$$

and for each of these ψ_e is the constant function e.

Define a partial recursive function θ by

$$\theta(e, x) = \begin{cases} \downarrow = 0 & \text{if } e = x \\ \uparrow & \text{otherwise} \end{cases}$$

By S-n-m theorem there is a recursive function g with $\psi_{g(e)}(x) = \theta(x)$. By the definition of θ we see that for every e:

$$W_{g(e)} = \{e\}$$

By the Recursion Theorem there are infinitely many fixed points for g and for any of them

$$W_e = W_{g(e)} = \{e\}.$$

Exercise 13.4. Prove:

(a) for every f, g recursive functions, there exists e_1 and e_2 such that $\psi_{f(e_1)} = \psi_{e_2}$ and $\psi_{g(e_2)} = \psi_{e_1}$

(b) $\exists e_1 \neq e_2 \quad W_{e_1} = \{e_2\},\ W_{e_2} = \{e_1\}$

(c) $\exists e_1 > e_2 > e_3 \quad W_{e_1} = \{e_2\},\ W_{e_2} = \{e_3\},\ W_{e_3} = \{e_1\}$

Exercise 13.5. Suppose $V \subseteq \omega$ is recursively enumerable. Show there exists infinitely many e such that $W_e = V_e$ where $V_e = \{n : \langle e, n\rangle \in V\}$.

Exercise 13.6. Prove there is a strictly increasing recursive function $f : \omega \to \omega$ such that $W_{f(n)} = \{n + f(n)\}$ for all n.

Example 13.7 *(Smullyan) For any recursive functions $f(x,y)$ and $g(x,y)$ there exists $a,b\in\omega$ such that*

$$\psi_{f(a,b)} = \psi_a \text{ and } \psi_{g(a,b)} = \psi_b$$

Proof
By the recursion theorem

$$\forall x\ \exists y\ \ \psi_{g(x,y)} = \psi_y$$

but since the fixed point y is obtained effectively from x and an index for g there exists a recursive function h such that

$$\forall x\ \ \psi_{g(x,h(x))} = \psi_{h(x)}$$

Apply the fixed point theorem to $f(x,h(x))$ there exists $a\in\omega$ such that

$$\psi_{f(a,h(a))} = \psi_a$$

Letting $b=h(a)$ does the job.
QED

Exercise 13.8. Prove
(a) $\exists e_1 < e_2 < e_3 \quad W_{e_1} = \{e_2\},\ W_{e_2} = \{e_3\},\ W_{e_3} = \{e_1\}$
(b) $\exists e_1 \neq e_2 \quad W_{e_1} = \{e_1, e_2\} = W_{e_2}$
(c) $\exists e_1 < e_2 < e_3 \quad W_{e_1} = \{e_2, e_3\},\ W_{e_2} = \{e_1, e_3\},\ W_{e_3} = \{e_1, e_2\}$

14 Myhill's characterization of creative set

Definition 14.1 *A r.e. set A is m-complete iff $B \leq_m A$ for every r.e. B. Similarly 1-complete.*

Definition 14.2 *An r.e. set C is creative iff there exists a recursive function $q\in\omega^\omega$ such that for every e*

$$W_e \cap C = \emptyset \rightarrow q(e) \notin C \cup W_e.$$

Theorem 14.3 *(Myhill) For $C\subseteq\omega$ r.e. the following are equivalent:*

1. *C is creative*

2. $C \equiv_1 K$

3. C *is 1-complete*

4. C *is m-complete*

Proof

(2) $\to$ (3): It is enough to see that K is 1-complete, since then for any B r.e. we would have $B \leq_1 K \leq_1 A$. Define a partial recursive function ρ as follows:

$$\rho(e,x) = \begin{cases} \downarrow = 0 & \text{if } e \in B \\ \uparrow & \text{otherwise} \end{cases}$$

ρ is partial recursive because we enumerate B looking to see if e ever turns up, if not the computation never halts. Using the 1-1-S-1-1 Theorem there exists a 1-1 recursive function f such that

$$\forall e, x \;\; \psi_{f(e)}(x) = \rho(e,x) = \begin{cases} \downarrow = 0 & \text{if } e \in B \\ \uparrow & \text{otherwise} \end{cases}$$

Then $e \in B$ iff $\psi_{f(e)}(f(e)) \downarrow$ iff $f(e) \in K$.

(3) $\to$ (4): Trivial

(4) $\to$ (1): The creativity of K is witnessed by the identity function, i.e.,

$$W_e \cap K = \emptyset \to e \notin W_e \cup K.$$

Suppose $K \leq_m A$ is witnessed by the function f. Then there exists a recursive function q such that

$$\text{for all } e \quad W_{q(e)} = f^{-1}(W_e)$$

(Use S-1-1 to get $\psi_{q(e)} = \psi_e \circ f$.) Then

$$W_e \cap A = \emptyset \to$$

$$f^{-1}(W_e) \cap K = \emptyset \to$$

$$W_{q(e)} \cap K = \emptyset \to$$

$$q(e) \notin f^{-1}(W_e) \cup K \to$$

$$f(q(e)) \notin W_e \cup A$$

so $f \circ q$ witnesses the creativity of A.

(1) $\to$ (2):

Claim The creativity function for A can be taken to be 1-1.

Proof

Given any creativity function d for A. Construct a recursive function f such that

$$\forall x \quad W_{f(x)} = W_x \cup \{d(x)\}.$$

To do this use

$$\forall x, y \;\; \psi_{f(x)}(y) = \rho(x,y) = \begin{cases} \downarrow = 0 & \text{if } y \in W_x \text{ or } y = d(x) \\ \uparrow & \text{otherwise} \end{cases}$$

Now we get a strictly increasing creativity function $\hat{d}$ recursively as follows: Input e put $e = e_0$ and effectively generate the sequence e_{s+1} where $W_{e_{s+1}} = W_{e_s} \cup \{d(e_s)\}$, i.e. put $e_{s+1} = f(e_s)$.

Search for the least s such that either

1. $d(e_s) > \hat{d}(e-1)$ or
2. $d(e_s) = d(e_t)$ for some $t < s$.

If the first happens put $\hat{d}(e) = d(e_s)$. If the second happens, then we know it is not the case that $W_e \subseteq \overline{A}$, because then W_{e_s} are all subsets of $\overline{A}$ and the $d(e_s)$ are all distinct. So in this case we may put $\hat{d}(e)$ to anything we like: e.g. put $\hat{d}(e) = \hat{d}(e-1) + 1$.

This proves the Claim.

QED

Now we show that $K \leq_1 A$. Define a partial recursive function θ as follows:

$$\psi_{f(n,x}(y) = \theta(n,x,y) = \begin{cases} \downarrow = 0 & \text{if } n \in K \text{ and } y = \hat{d}(x) \\ \uparrow & \text{otherwise} \end{cases}$$

It follows that

$$W_{f(n,x)} = \begin{cases} \{\hat{d}(x)\} & \text{if } n \in K \\ \emptyset & \text{otherwise} \end{cases}$$

By the uniform proof of the recursion theorem and by padding we get a 1-1 recursive sequence $n \mapsto e_n$ of fixed points so that

$$\forall n \;\; W_{f(n,e_n)} = W_{e_n} = \begin{cases} \{\hat{d}(e_n)\} & \text{if } n \in K \\ \emptyset & \text{otherwise} \end{cases}$$

But then $n \in K$ iff $\hat{d}(e_n) \in A$. So $K \leq_1 A$.
QED

Most naturally occurring nonrecursive r.e. sets are m-complete.

Exercise 14.4. Prove or disprove: there exists a creative set A and a recursive function $q : \omega \to \omega$ such that for every e

$$W_e \cap A \text{ finite } \to q(e) \notin W_e \cup A.$$

Exercise 14.5. Prove that a r.e. set A is creative iff there exists a computable f such that for every e

1. $W_e \cap A = \emptyset \to f(e) \notin W_e \cup A$ and
2. $W_e \cap A \neq \emptyset \to f(e) \in W_e \cap A$.

15 Simple sets

Definition 15.1 *A is simple iff A is r.e. , $\overline{A}$ is infinite, and $\overline{A}$ does not contain an infinite r.e. set.*

Theorem 15.2 *(Post) There exists a simple set.*

Proof
Define a recursive sequence $A_s \subseteq s$ of increasing finite sets as follows. $A_0 = \emptyset$. At stage $s+1$ find the least $e < s$ (if any) such that $W_{e,s} \cap A_s = \emptyset$ and $\exists x > 2e\ x \in W_{e,s}$. Put $A_{s+1} = A_s \cup \{x\}$ for the least e and x for which this is true. If this happens we say that e has acted at stage $s+1$. If there no such e, then put $A_{s+1} = A_s$.

The set $A = \cup_s A_s$ is simple. Note that each e can act at most once. Hence if W_e is infinite and $W_e \cap A = \emptyset$, eventually there will come a stage s where $\exists x > 2e\ x \in W_{e,s}$ and all smaller e's which will ever act have already acted at a previous stage. But then e will act, which is a contradiction.

Also we see that $\overline{A}$ is infinite because for all e $|A \cap 2e| \leq e$ since the only e' which can put an x into A with $x \leq 2e$ are those e' with $e' < e$.
QED

Exercise 15.3. Are there always recursive Skolem functions? Prove or disprove:

(a) Given a recursive $R \subseteq \omega^2$ such that $\forall x \exists y\ R(x,y)$ there exists a recursive f such that $\forall x\ R(x, f(x))$

(b) Given a recursive $R \subseteq \omega^3$ such that $\forall x \exists y \forall z\ R(x,y,z)$ there exists a recursive f such that $\forall x \forall z\ R(x, f(x), z)$

Hint: Think "Simple".

Exercise 15.4. Suppose A is a simple set and $A = \{a_n : n \in \omega\}$ is a 1-1 recursive enumeration of A. Prove there exists infinitely many n such that $W_{a_n} = \{a_m : m > n\}$. (Hint: it is easier to show there exists $e \in A$ such that $W_e = \{e\}$.)

Exercise 15.5 *Show that*

(a) If $A \leq_1 B$ and B is simple, then A is simple or $\overline{A}$ is finite.

(b) If A and B are simple, then $A \cup B$ is simple.

(c) If A is simple, $b \in \overline{A}$, and $B = A \cup \{b\}$, then $B <_1 A$ and if $B \leq_1 C \leq_1 A$ then $C \equiv_1 B$ or $C \equiv_1 A$.

16 Oracles

Definition 16.1 *$A \leq_T B$ or A is Turing reducible to B. Add to the UR-Basic programming language statements of the form:*

$$\text{Let } y = \chi_B(x)$$

for any variables x, y. This programming language is called Oracle UR-Basic. Then $A \leq_T B$ iff there is an Oracle UR-Basic program with Oracle for B which computes the characteristic function χ_A of A.

17 Dekker deficiency set

Proposition 17.1 *(Dekker Deficiency Set) For every r.e. set A which is not recursive there exists a simple set B with $B \equiv_T A$.*

Proof
Let $\{a_n : n \in \omega\}$ be a 1-1 recursive enumeration of A. Define

$$B = \{n : \exists m > n\ \ a_m < a_n\}$$

It is easy to see that B is r.e.

$\overline{B}$ is infinite: Otherwise there would be an N such that $a_{n+1} > a_n$ for all $n > N$ and then A would be recursive.

$A \leq_T B$: Input x. Find $n \in \overline{B}$ such that $a_n > x$. Then $x \in A$ iff $x \in \{a_i : i < n\}$.

$\overline{B}$ does not contain an infinite recursive set: Suppose $R \subseteq \overline{B}$ is an infinite recursive set. But then the argument we just gave for $A \leq_T B$ shows that $A \leq_T R$ which would make A recursive.

$B \leq_T A$: Input n. Using an Oracle for A check if

$$\{a_i \ : \ a_i < a_n \text{ and } i < n\} = A \cap \{x \ : \ x < a_n\}$$

if they are equal, then $n \notin B$, otherwise $n \in B$.
QED

Exercise 17.2. (From Cooper) Define $B \subseteq \omega$ is intro-reducible iff $B \leq_T C$ for every infinite $C \subseteq B$. Prove that for every A there exists $B \equiv_T A$ intro-reducible.

18 Turing degrees and jumps

Definition 18.1 *For $A \subseteq \omega$ define the Turing degree of A to be*

$$a = deg(A) = \{B \ : \ B \equiv_T A\}.$$

Let $\mathcal{D} = \{deg(A) \ : \ A \subseteq \omega\}$ be the Turing Degrees. $(\mathcal{D}, \leq)$ is the partial order where $a \leq b$ iff $A \leq_T B$.

Definition 18.2 *For $\sigma \in 2^{<\omega}$ and $e, x, y, s \in \omega$ we write*

$$\{e\}_s^\sigma(x) \downarrow = y$$

to mean that the e^{th} oracle machine with input x and using σ to answer Oracle questions, converges in less than s steps and outputs y. We also require that $e, x, y < s$ and that in this computation the oracle is not asked about any n such that $n \notin dom(\sigma)$ or $n \geq s$.

Proposition 18.3 *The predicate $O(\sigma, e, x, y, s)$ defined by*

$$O(\sigma, e, x, y, s) \text{ iff } \{e\}_s^\sigma(x) \downarrow = y$$

is primitive recursive.

Definition 18.4 *For $A \subseteq \omega$ the jump of A is defined by*

$$A' = \{e : \exists s \;\; e_s^{A \restriction s}(e) \downarrow\}$$

Proposition 18.5 *(1) $A \leq_T B$ implies $A' \leq_1 B'$.*
(2) $A <_T A'$

Proof
(1) Define

$$\theta(e,x) = \begin{cases} \downarrow = 0 & \text{if } e^A(e)\downarrow \\ \uparrow & \text{otherwise} \end{cases}$$

Then θ is partial recursive in A and since $A \leq_T B$ we have that θ is partial recursive in B. By the 1-1-S-1-1 Theorem relativized to B there exists a 1-1 recursive function f such that

$$\forall e, x \;\; \{f(e)\}^B(x) = \theta(e,x).$$

But then $e \in A'$ iff $\{e\}^A(e)\downarrow$ iff $\{f(e)\}^B(f(e))\downarrow$ iff $f(e) \in B'$.

(2) To see $A \leq_1 A'$ construct a 1-1 recursive function f so that $\{f(n)\}^A(?)$ has the same computation on any input and it converges iff $n \in A$. Then $n \in A$ iff $f(n) \in A'$. To see that $A' \not\leq_T A$, suppose that it is. Define $f = 1 - \chi_{A'}$. Then since $f \leq_T A' \leq_T A$ there is an e_0 with $\{e_0\}^A = f$. But then $e_0 \in A'$ iff $e_0 \notin A'$.
QED

Corollary 18.6 *If $A \equiv_T B$, then $A' \equiv_T B'$. Hence, letting $a' \in \mathcal{D}$ be the Turing degree of A' is well-defined and $a < a'$ for every $a \in \mathcal{D}$.*

Similarly, a'' is the jump of the jump of a, and $a^{(n)}$ is n jumps of a.

19 Kleene-Post: incomparable degrees

Definition 19.1 *$a|b$ iff not $a \leq b$ and not $b \leq a$. I.e. the degrees a and b are Turing incomparable.*

Proposition 19.2 *(Kleene-Post) There exists $a, b \in \mathcal{D}$ with $a|b$.*

Proof

Construct sequences $(\sigma_s \in 2^{<\omega} : s \in \omega)$, $(\tau_s \in 2^{<\omega} : s \in \omega)$ with the property that $\sigma_s \subseteq \sigma_{s+1}$ and $\tau_s \subseteq \tau_{s+1}$ for each s. For $s = 0$ take τ_s and σ_s to be the empty sequence.

At stage $s + 1$ we are given τ_s and σ_s and we do as follows:

Case $s = 2e$:

Let $n = |\tau_s|$.

Case a. There exists $\sigma \supseteq \sigma_s$ such that $\{e\}^\sigma(n)\downarrow$. In this case put $\sigma_{s+1} = \sigma$ and put $\tau_{s+1} = \tau_s i$ where $i = 0, 1$ whichever is different from $\{e\}^\sigma(n)$.

Case b. No such σ. Put $\sigma_{s+1} = \sigma_s$ and $\tau_{s+1} = \tau_s 0$.

Case $s = 2e + 1$:

Let $n = |\sigma_s|$ and proceed similarly to $s = 2e$ with the roles of σ_s and τ_s reversed.

This ends the construction. We put $A = \cup_{s\in\omega}\sigma_s$ and $B = \cup_{s\in\omega}\tau_s$.

QED

It is easy to see that the entire construction is recursive in o' and hence there are incomparable Turing degrees beneath o'.

Proposition 19.3 *(Kleene-Post) For every $a \in \mathcal{D} \setminus \{o\}$ there exists $b \in \mathcal{D}$ with $a|b$.*

Let $deg(A) = a$. Construct $(\tau_s \in 2^{<\omega} : s \in \omega)$ as follows. $\tau_0 = \langle\rangle$.

At stage $s + 1$ we are given τ_s.

Case $s = 2e$. Let $n = |\tau_s|$. Take $i = 0$ or $i = 1$ so that $i \neq \{e\}^A(n)$. Put $\tau_{s+1} = \tau_s i$.

Case $s = 2e + 1$.

Case a. There exists $n < \omega$, ρ_1, ρ_2 with $\tau_s \subseteq \rho_i$ and

$$\{e\}^{\rho_1}(n)\downarrow \neq \{e\}^{\rho_2}(n)\downarrow$$

In this case we put $\tau_{s+1} = \rho_1$ or $\tau_{s+1} = \rho_2$ which ever that case is that

$$\{e\}^{\tau_{s+1}}(n) \neq A(n).$$

Case b. There is no such n and ρ_i. Put $\tau_{s+1} = \tau_s 0$.

This ends the construction. Now we check that $B = \cup_s \tau_s$ is Turing incomparable to A. The cases $2e$ easily show that $B \not\leq_T A$. Suppose $A \leq_T B$ and choose e so that $\{e\}^B = A$ and consider stage $s+1$ where $s = 2e+1$. In case (a) we get that $\{e\}^B(n) \neq A(n)$ so that it is impossible. Now we show that case (b) cannot happen. Define

$$f(n) = i \text{ iff } \exists \tau \supseteq \tau_s \{e\}^\tau(n) \downarrow = i$$

Note that f is well-defined because we are in case (b) and f is total because we are assume that $\{e\}^B$ is the characteristic function of A. Hence f which is recursive is the characteristic function of A, which contradicts the assumption that A is not recursive.
QED

Exercise 19.4. Prove that for every countable $\mathcal{A} \subseteq \mathcal{D} \setminus \{0\}$ there exists $b \in \mathcal{D}$ such that $a|b$ for all $a \in \mathcal{A}$.

20 The join

Definition 20.1 $A \oplus B = \{2n \ : \ n \in A\} \cup \{2n+1 \ : \ n \in B\}$.

Exercise 20.2. Prove
(a) $A \leq_T A \oplus B$ and $B \leq_T A \oplus B$
(b) $A \oplus B \equiv_T B \oplus A$
(c) $(A \oplus B) \oplus C \equiv_T A \oplus (B \oplus C)$
(d) if $A \leq_T C$ and $B \leq_T C$, then $A \oplus B \leq_T C$
(e) if $A \leq_T \hat{A}$ and $B \leq_T \hat{B}$, then $A \oplus B \leq_T \hat{A} \oplus \hat{B}$

Definition 20.3 $a \vee b = deg(A \oplus B)$ *is the join or least upper bound of* a *and* b.

Exercise 20.4 *Show that if* A *and* B *are simple, then* $A \oplus B$ *is simple.*

Exercise 20.5. (Young) Suppose A and B are simple and are $\leq_1$ incomparable. Prove that they have no join with respect to $\leq_1$. That is, there is no C such

1. $A \leq_1 C$ and $B \leq_1 C$ and
2. for all D if $A \leq_1 D$ and $B \leq_1 D$, then $C \leq_1 D$.

Note that $A \oplus B$ does not work and nothing else does either. Hint: Use exercises 20.4, 15.5, and 9.4.

21 Meets

Meets, $a \wedge b$, in the Turing degrees may or may not exist.

Proposition 21.1 *(Kleene-Post) There exists $a, b \in \mathcal{D} \setminus \{o\}$ with $a \wedge b = 0$ i.e., for all c if $c \leq a$ and $c \leq b$ then $c = o$.*

Proof
As before construct sequences $(\sigma_s \in 2^{<\omega} : s \in \omega)$, $(\tau_s \in 2^{<\omega} : s \in \omega)$ with the property that $\sigma_s \subseteq \sigma_{s+1}$ and $\tau_s \subseteq \tau_{s+1}$ for each s. For $s = 0$ take τ_s and σ_s to be the empty sequence.

At stage $s + 1$ we are given τ_s and σ_s and we do as follows:

Case $s = 3e$. Let $n = |\sigma_s|$. Let $i = 0$ or $i = 1$ so that $\psi_e(n) \neq i$. Put $\sigma_{s+1} = \sigma_s i$.

Case $s = 3e + 1$. Similar to $3e$ but for τ_{s+1}.

Case $s = 3\langle e_1, e_2\rangle + 2$.

Case a. There exists $n < \omega$, $\sigma \supseteq \sigma_s$, and $\tau \supseteq \tau_s$ such that

$$\{e_1\}^\sigma(n) \downarrow \neq \{e_2\}^\tau(n) \downarrow$$

put $\sigma_{s+1} = \sigma$ and $\tau_{s+1} = \tau$.

Case b. Not case a. Put $\tau_{s+1} = \tau_s$ and $\sigma_{s+1} = \sigma_s$.

This ends the construction. We put $A = \cup_s \sigma_s$ and $B = \cup_s \tau_s$. The stages $3e, 3e + 1$ guarantee that neither A nor B is recursive. Now suppose that $C \leq_T A$ and $C \leq_T B$. This will be witnessed by a pair e_1 and e_2. At stage $s = 3\langle e_1, e_2\rangle + 2$ it must have been that Case a. failed since we assume that

$$\{e_1\}^A = \{e_2\}^B = C.$$

But then we may define a total recursive function f by

$$f(n) = i \text{ iff } \exists \sigma \supseteq \sigma_s \ \ \{e_1\}^\sigma(n) \downarrow= i$$

and f must be the characteristic function of C and hence C is recursive.
QED

Proposition 21.2 *(Kleene-Post) For every $c \in \mathcal{D}$ there exists $a, b \in \mathcal{D}$ with $a \wedge b = c$ and $a|b$, i.e., $a > c$, $b > c$, and for all d if $d \leq a$ and $d \leq b$ then $d \leq c$.*

Proof
This is a relativization of the above argument. Construct A_0 and B_0 so that for every e

$$\{e\}^C \neq A_0 \oplus C \text{ and } \{e\}^C \neq B_0 \oplus C$$

and

$$\{e_1\}^{A_0 \oplus C} = \{e_2\}^{B_0 \oplus C} = D \rightarrow D \leq_T C$$

Then take $A = A_0 \oplus C$ and $B = B_0 \oplus C$.
QED

Exercise 21.3. Find a minimal triple, i.e., $a, b, c \in \mathcal{D} \setminus \{0\}$ such that

$$\forall d \ \ (d \leq a \text{ and } d \leq b \text{ and } d \leq c) \rightarrow d = 0$$

but no 2 are a minimal pair.

Hint: Construct X, Y, Z non recursive so that

$$(\{e_0\}^{X \oplus Y} = \{e_1\}^{Y \oplus Z} = \{e_2\}^{X \oplus Z} = D) \rightarrow D \leq_T 0.$$

Exercise 21.4. Prove:

(a) There exists $A \subseteq \omega$ such that $A_n \not\leq_T \hat{A}_n$ for every n where

$$A_n = \{x : \langle n, x\rangle \in A\} \text{ and } \hat{A}_n = \{\langle m, x\rangle : m \neq n \text{ and } \langle m, x\rangle \in A\}.$$

(b) There exists Turing degrees a_r for $r \in \mathbf{Q}$ such that for all $r, s \in \mathbf{Q}$ ($r < s$ iff $a_r < a_s$). Hint: use part (a).

(c)* Same as part (b) but also $a_r < 0'$ for all r.

Exercise 21.5. Prove that for every $b \in \mathcal{D}$ with $b > o$ there exists $a \in \mathcal{D}$ with $a > o$ and $a \wedge b = 0$.

Exercise 21.6. Prove that for every $c \in \mathcal{D}$ with $c \geq o'$ that there exists incomparable degrees a and b with $a \wedge b = 0$, and $a \vee b = c$.

Hint: one way to code a set C into $A \oplus B$ is to use boot-strapping. Define

$$x_{2n} = \mu x > x_{2n-1}\ A(x) = 1$$

$$x_{2n+1} = \mu x > x_{2n}\ B(x) = 1$$

$$n \in C \text{ iff } x_n \text{ is even.}$$

22 Spector: exact pairs

Proposition 22.1 *(Spector) Given $(a_n : n < \omega)$ in $\mathcal{D}$ with $a_n < a_{n+1}$ for all n there exists $b, c \in \mathcal{D}$ with*

(1) $a_n \leq b$ and $a_n \leq c$ for all n and

(2) for all $d \in \mathcal{D}$ if $d \leq b$ and $d \leq c$ then there exists n with $d \leq a_n$.

Proof

Let $deg(A_n) = a_n$ and set $A = \{\langle n, x\rangle : n < \omega, x \in A_n\}$. The key to this construction is to make B and C have the property that for each n

$$B_n =^* A_n =^* C_n$$

where $B_n = \{x : \langle n, x\rangle \in B\}$ and $C_n = \{x : \langle n, x\rangle \in C\}$. The symbol $X =^* Y$ means "equal except for a finite set".

As before construct sequences $(\sigma_s \in 2^{<\omega} : s \in \omega)$, $(\tau_s \in 2^{<\omega} : s \in \omega)$ with the property that $\sigma_s \subseteq \sigma_{s+1}$ and $\tau_s \subseteq \tau_{s+1}$ for each s. For $s = 0$ take τ_s and σ_s to be the empty sequence.

At stage $s + 1$ we will extend σ_s and τ_s so as to agree with A_i for $i < s$ on new elements of their domain. Define

$$f_s = \sigma_s \cup \{\langle\langle i, x\rangle, j\rangle : \langle i, x\rangle \notin dom(\sigma_s),\ \ i < s,\ \text{and}\ A_i(x) = j\}$$

$$g_s = \tau_s \cup \{\langle\langle i, x\rangle, j\rangle : \langle i, x\rangle \notin dom(\tau_s),\ \ i < s,\ \text{and}\ A_i(x) = j\}$$

Note that f_s is a partial function extending σ_s which agrees with the characteristic function of each A_i for $i < s$ except possible on the (finite) domain of σ_s. Similarly g_s.

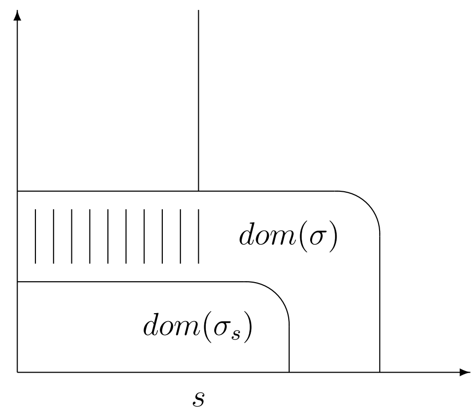


Figure 4: σ must agree with A on the shaded region.

Let $s = \langle e_1, e_2 \rangle$.

Case a. There exists $n < \omega$, $\sigma \supseteq \sigma_s$ and $\tau \supseteq \tau_s$ such that $f_s \cup \sigma$ is a function (i.e., they are compatible - see Figure 4) and $g_s \cup \tau$ is a function and

$$\{e_1\}^\sigma(n) \downarrow \neq \{e_2\}^\tau(n) \downarrow .$$

Put $\sigma_{s+1} = \sigma$ and $\tau_{s+1} = \tau$.

Case b. Not Case a. Put $\sigma_{s+1} = \sigma_s$ and $\tau_{s+1} = \tau_s$.

This completes the construction, so put $B = \cup_s \sigma_s$ and $C = \cup_s \tau_s$.

Claim. For all n we have that $A_n \leq_T B$ and $A_n \leq_T C$. To see this note that in the construction that for all $s > n$ that $f_s(\langle n, m \rangle) = f_{n+1}(\langle n, m \rangle)$. Furthermore, except for the finitely many element of the domain of σ_{n+1} we have that $A_n(m) = f_{n+1}(\langle n, m \rangle)$. It follows that $A_n =^* B_n$ and so $A_n \leq_T B_n \leq_T B$. Similarly for C.

Claim. Suppose that $D \leq_T B$ and $D \leq_T C$. Then $D \leq_T A_n$ for some $n < \omega$. To see this suppose that

$$\{e_1\}^B = \{e_2\}^C = D$$

and $s = \langle e_1, e_2 \rangle$. Since the characteristic functions of B and C extend σ_{s+1} and τ_{s+1} respectively it is evident that Case (a) could not have occurred. So we assume Case (b). Note that in this case it is impossible that there exists n, ρ_1, ρ_2 with $\sigma_s \subseteq \rho_1$ and $\sigma_s \subseteq \rho_2$, and each of ρ_1 and ρ_2 compatible with f_s such that

$$\{e_1\}^{\rho_1}(n) \downarrow \neq \{e_1\}^{\rho_2}(n) \downarrow .$$

This is because $\{e_2\}^C(n) \downarrow$ and so then we would be in Case (a).

It follows easily as before that $D = \{e_1\}^B \leq_T f_s$. But

$$f_s \leq_T A_0 \oplus A_1 \oplus \cdot \oplus A_{s-1} \leq_t A_{s-1}$$

so $D \leq_T A_{s-1}$.
QED

Exercise 22.2. Suppose $a, b \in \mathcal{D}$ and $a \wedge b$ does not exist. Prove there exists $(c_n \in \mathcal{D} : n < \omega)$ such that

1. $c_n \leq a$ and $c_n \leq b$ for all n,
2. $c_n < c_{n+1}$ for all n, and
3. for all $d \in \mathcal{D}$ if $d \leq a$ and $d \leq b$, then $d \leq c_n$ for some n.

23 Friedberg: jump inversion

Proposition 23.1 *(Friedberg Jump Inversion) For every $a \in \mathcal{D}$ if $a \geq o'$ then there exists $b \in \mathcal{D}$ with $b' = a$.*

Proof
We construct sequence $(\tau_s : s \in \omega)$ recursive in $A \oplus 0' \equiv_T A$ as follows.

At stage $s+1$ we are given $\tau_s \in 2^{<\omega}$

(a) We put $\tau = \tau_s i$ where $i = A(s)$.

(b) Let $e = s$. We ask $0'$ if there exists $\sigma \supseteq \tau$ such that

$$\{e\}^{\sigma}_{|\sigma|}(e) \downarrow$$

If there is such a σ then we effectively find one and put $\tau_{s+1} = \sigma$.

More precisely, before the construction begins find a recursive function $f(e, \tau)$ such that

1. for any e, τ

$$\psi_{f(e,\tau)}(0) \downarrow \text{ iff } \exists \sigma \supseteq \tau \;\; \{e\}^{\sigma}_{|\sigma|}(e) \downarrow$$

2. when $\psi_{f(e,\tau)}(0)$ converges it outputs such a σ and

3. the algorithm $\psi_{f(e,\tau)}(?)$ ignores its input.

We put $\tau_{s+1} = \tau$ if $f(e, \tau) \notin 0'$, otherwise we put $\tau_{s+1} = \sigma =^{def} \psi_{f(e,\tau)}(0)$.

This ends the construction. We let $B = \cup_{s\in\omega}\tau_s$.

Claim.

1. $(\tau_s : s \in \omega) \leq_T A \oplus 0' \leq_T A$

2. $A \leq_T (\tau_s : s \in \omega)$

3. $(\tau_s : s \in \omega) \leq_T B \oplus 0'$

4. $B' \leq_T (\tau_s : s \in \omega)$

Proof

(1) The construction only requires oracles for $0'$ and A. Also $A \geq_T 0'$.

(2) We encoded the characteristic function of A at step (a). Hence

$$s \in A \text{ iff } \tau_{s+1}(|\tau_s|) = 1.$$

(3) Recursively construct the sequence $(\tau_s : s \in \omega)$ using oracles for $0'$ and B. Given τ_s we use that $\tau_{s+1} \subseteq B$ to figure out the first digit, i.e., τ of step (a). To do step (b) we only used $0'$ and the recursive function f.

(4) By our construction given any e let $s = e$, then we have that

$$e \in B' \text{ iff } \{e\}^B(e)\downarrow \text{ iff } \{e\}^{\tau_{s+1}}_{|\tau_{s+1}|}(e)\downarrow$$

This proves the Claim. But note that the Claim implies

$$B' \leq_T (\tau_s : s \in \omega) \leq_T A \leq_T (\tau_s : s \in \omega) \leq_T B \oplus 0' \leq_T B'$$

QED

Exercise 23.2. Prove that $\forall a \in \mathcal{D} \quad a \geq o' \rightarrow \exists b, c \in \mathcal{D} \quad b|c$ and $b' = a = c'$.

24 Spector: minimal degree

Theorem 24.1 *(Clifford Spector) There exists a minimal Turing degree, i.e., $\exists a \in \mathcal{D}$ with $o < a$ but no $b \in \mathcal{D}$ with $o < b < a$.*

Proof
For any $\sigma \in 2^n$, i.e., a finite sequence of zeros and ones, we can code σ by the number

$$x = 2^n + \sum\{2^i \ : \ i < n \text{ and } \sigma(i) = 1\}.$$

The extra 2^n is there to distinguish sequences ending in zeros from each other. We suppress this coding and just talk about recursive subsets of $2^{<\omega}$.

Definition 24.2 *$T \subseteq 2^{<\omega}$ is a perfect tree iff*

1. *T is nonempty,*
2. *$\sigma \subseteq \tau \in T$ implies $\sigma \in T$, and*
3. *$\forall \sigma \in T \ \ \exists \tau_0, \tau_1 \in T$ with $\sigma \subseteq \tau_0$, $\sigma \subseteq \tau_1$, and τ_0 and τ_1 are incomparable.*

Definition 24.3 *For $T \subseteq 2^{<\omega}$ a tree we define:*

1. *$\sigma \in T$ splits iff $\sigma 0, \sigma 1 \in T$*
2. *$\sigma = stem(T)$ iff σ splits but no shorter node of T splits*
3. *$[T] = \{x \in 2^\omega \ : \ \forall n \ \ x{\restriction}n \in T\}$*
4. *for $\sigma \in T$ let*
$$T(\sigma) = \{\tau \in T \ : \ \tau \subseteq \sigma \text{ or } \sigma \subseteq \tau\}$$

To prove the Theorem construct a sequence $(T_s \ : \ s \in \omega)$ of recursive perfect trees as follows.

At stage $s = 0$ take $T_0 = 2^{<\omega}$.

At stage $s+1$ where $s = 2e$ let $\sigma = stem(T_s)$ and $n = |\sigma|$. If $\psi_e(n) \downarrow = 0$ then put $T_{s+1} = T_s(\sigma 1)$ otherwise put $T_{s+1} = T_s(\sigma 0)$.

At stage $s+1$ where $s = 2e+1$ we obtain $T_{s+1} \subseteq T_s$ a perfect recursive subtree as follows. We first ask the question:

> Does there exist $\sigma \in T_s$ such that for all $\sigma_1, \sigma_2 \in T(\sigma)$ and $n, m_1, m_2 < \omega$ if $\{e\}^{\sigma_1}(n) \downarrow = m_1$ and $\{e\}^{\sigma_2}(n) \downarrow = m_2$, then $m_1 = m_2$?

Case (a) If the answer is yes, we take $T_{s+1} = T_s(\sigma)$ for any such σ.
Case (b) If the answer is no, we construct recursive sequences
$(\sigma_\rho \in T : \rho \in 2^{<\omega})$ and $(n_\rho \in \omega : \rho \in 2^{<\omega})$
such that

1. $\{e\}^{\sigma_{\rho 0}}(n_\rho) \downarrow \neq \{e\}^{\sigma_{\rho 1}}(n_\rho) \downarrow$ and
2. $\sigma_\rho \subseteq \sigma_{\rho 0}$ and $\sigma_\rho \subseteq \sigma_{\rho 1}$.

Note that (1) implies that $\sigma_{\rho 0}$ is incomparable to $\sigma_{\rho 1}$. We put

$$T_{s+1} = \{\sigma \ : \ \exists \rho \in 2^{<\omega} \ \ \sigma \subseteq \sigma_\rho\}$$

then T_{s+1} is a recursive perfect subtree of T_s.

This ends the construction of the sequence of trees. Note that $T_{s+1} \subseteq T_s$. Take A to be the subset of ω whose characteristic function is the unique element of $\cap_{s \in \omega}[T_s]$. It is easy to see that stage $2e+1$ guarantees that A is not recursive, so it is enough to see stage $2e+2$ guarantees that if $B = \{e\}^A$ then either B is recursive or $A \leq_T B$.

Case (a) for all $\sigma_1, \sigma_2 \in T_{s+1}$ and $n, m_1, m_2 < \omega$ if $\{e\}^{\sigma_1}(n) \downarrow = m_1$ and $\{e\}^{\sigma_2}(n) \downarrow = m_2$, then $m_1 = m_2$. In this case B is recursive, since $A \in [T_{s+1}]$ and $B = \{e\}^A$ means that all we have to do to compute $B(n)$ is to search the recursive tree T_{s+1} for any σ for which $\{e\}^\sigma(n) \downarrow$ and then $B(n) = \{e\}^\sigma(n)$.

Case (b) In this case we show that $A \leq_T B$. We know $A \in [T_{s+1}]$. Suppose we know that $\sigma_\rho \subseteq A$. To decide whether $\sigma_{\rho 0} \subseteq A$ or $\sigma_{\rho 1} \subseteq A$, we compute both of

$$\{e\}^{\sigma_{\rho 0}}(n_\rho) \text{ and } \{e\}^{\sigma_{\rho 1}}(n_\rho).$$

Since these two computations are guaranteed to converge and to different values at most one of them can agree with $B(n_\rho)$. One of them must agree and so using an oracle for B we can determine the unique $i = 0, 1$ so that $\sigma_{\rho i} \subseteq A$.
QED

Exercise 24.4. Prove that there are uncountably many minimal degrees.

Exercise 24.5. Prove there exists a perfect tree $T \subseteq 2^{<\omega}$ such that for every n and distinct $y, x_1, x_2, \ldots, x_n \in [T]$

$$y \not\leq_T x_1 \oplus x_2 \oplus \cdots \oplus x_n.$$

25 Sacks: minimal upper bounds

Theorem 25.1 *(Sacks) Minimal upper bounds exists. Given any sequence of degrees* $(a_n \in \mathcal{D} : n < \omega)$ *such that* $a_n < a_{n+1}$ *for all* n *there exists* $b \in \mathcal{D}$ *with* $a_n < b$ *all* n *but there is no* $c \in \mathcal{D}$ *with* $a_n < c < b$ *for all* n.

Proof
Here we use the notion of a recursively-pointed tree.

Definition 25.2 $T \subseteq 2^{<\omega}$ *is recursively-pointed iff* T *is a perfect tree and* $T \leq_T A$ *for every* $A \in [T]$.

The new ingredient required in this construction is

Claim. Suppose $T \subseteq 2^{<\omega}$ is recursively-pointed tree and $T \leq_T B$. Then there exists $T^* \subseteq T$ a recursively-pointed tree such that $T^* \equiv_T B$.
Proof
There exists a natural bijection $f : 2^{<\omega} \to Split(T)$ where $Split(T)$ are the splitting nodes of T. Note that f and T are Turing equivalent. Given $B \in 2^{\omega}$ let

$$T_B = \{\sigma \in 2^{<\omega} : \sigma(2n) = B(n) \text{ whenever } 2n < |\sigma|\}.$$

Now take T^* to be the tree generated by $f(T_B)$.
QED

Construct $(T_s : s \in \omega)$ a sequence of recursively-pointed trees as follows.

Suppose $T_s \equiv_T A_s$ and $e = s$. Relativizing Spector's proof above to T_s we can obtain $T^\circ \subseteq T_s$ with $T^\circ \leq_T T_s$ a perfect subtree so that for every $B \in [T^\circ]$: if $C = \{e\}^B$ then either $B \leq_T (C \oplus T^\circ)$ or $C \leq_T T^\circ$.

Note that T° is recursively-pointed and $T^\circ \leq_T A_s$. Hence by applying the Claim above we can obtain $T_{s+1} \subseteq T^\circ$ such that T_{s+1} is recursively-pointed and $T_{s+1} \equiv_T A_{s+1}$.

This ends the construction. We let B be the unique element of $\cap_{s\in\omega}[T_s]$.

First note that $A_s \leq_T B$ for each s, because $B \in [T_s]$, T_s is recursively-pointed and so $A_s \equiv_T T_s \leq_T B$.

Suppose that $A_s \leq_T C \leq_T B$ for every $s \in \omega$. Then at some stage $s = e$ we have that $C = \{e\}^B$. Hence by construction either $C \leq_T T^\circ \leq_T A_s$ or $B \leq_T (C \oplus T^\circ)$. The first is impossible since $A_s <_T A_{s+1} \leq_T C$ and so it must be that $B \leq_T (C \oplus T^\circ)$. But $T^\circ \leq_T A_s \leq_T C$ so $B \leq_T C$.
QED

Exercise 25.3. (a) Prove there exists $a, b \in \mathcal{D}$ with $o < a < b$ and not there exists c with either $o < c < a$ or $a < c < b$.

(b) (Extra Credit) Prove there exists $a, b \in \mathcal{D}$ with $o < a < b$ and ($c \leq b$ iff $c = 0$ or $c = a$ or $c = b$), for all $c \in \mathcal{D}$.

Exercise 25.4. Show that the degree of

$$0^{(\omega)} = \{\langle n, x\rangle : x \in 0^{(n)}\}$$

is not a minimal upper bound of the degrees of $\{0^{(n)} : n \in \omega\}$.
Hint: in Theorem 22.1 get B, C computable in $0^{(\omega)}$.

Show there is $A \subseteq \omega$ such that for all n

$$0^{(n)} \leq_T A <_T A' \leq_T 0^{(\omega)}.$$

26 Friedberg-Muchnik Theorem

Definition 26.1 *The use of an oracle computation* $\{e\}^A(x)$ *written*

$$use(\{e\}^A(x))$$

is $n+1$ *where* n *is the maximum number for which the oracle for* A *is queried.*

Note that if $u = use(\{e\}^A(x))$ and $B \cap u = A \cap u$ then $\{e\}^A(x)$ and $\{e\}^B(x)$ are the same computation.

Theorem 26.2 *(Friedberg-Muchnik) There exists r.e. sets* A_0 *and* A_1 *such that* $A_0 \not\leq_T A_1$ *and* $A_1 \not\leq_T A_0$.

Proof
Our requirements are:

R_{2e+i} $\quad \{e\}^{A_i} \neq A_{1-i}$

for each $e \in \omega$ and $i = 0, 1$.

The strategy for meeting this requirement is to attach a follower $x \in \omega$ to R_{2e+i} and then wait until $\{e\}_s^{A_{i,s}}(x)\downarrow= 0$. When this happens we put x into A_{1-i} and try to avoid injuring the computation $\{e\}_s^{A_{i,s}}(x)$. If we succeed then $\{e\}^{A_i}(x) = 0 \neq 1 = A_{1-i}(x)$. If we wait forever, then x is never put into A_{1-i} and so $A_{1-i}(x) = 0 \neq \{e\}^{A_i}(x)$. In either case the requirement R_{2e+i} is met. There are two possible successful outcomes for this strategy, either we wait forever or we act at some stage and then preserved the relevant computation.

Construction

Everything in the construction will be done effectively.

At each stage s of the construction we will have effectively constructed:

1. finite sets $A_{i,s}$ for $i = 0, 1$,
2. a follower $x = x_{q,s}$ for each R_q with $q < s$, and
3. a function f_s with domain s which is attempting to predicate the final outcomes of our strategy for each R_q with $q < s$.

At stage $s = 0$ put $A_{i,0} = \emptyset$ for $i = 0, 1$. Nobody has followers and f_s is the empty function.

At stage $s + 1$ look for the least $q = 2e + i < s$ such that

1. $f_s(q) =$'waiting' and
2. $\{e\}_s^{A_{i,s}}(x)\downarrow= 0$ with use less than s where $x = x_{q,s}$ is the follower of R_{2e+i}.

If we find such a q then we take the following actions:

1. Put x into A_{1-i}, i.e.,
$$A_{1-i,s+1} = A_{1-i,s} \cup \{x\}$$
2. Set $f_{s+1}(q) =$'acted'.
3. Reappoint followers for lower priority requirements, i.e. for each $q' > q$ with $q' < s + 1$ put $x = \langle q', s + 1\rangle$ to be the follower of $R_{q'}$.
4. Make all lower priority requirements start over, i.e., for each $q' > q$ put $f_{s+1}(q') =$'waiting'.

We say that R_q acted at stage $s+1$. If there is no such q then we just continue to wait. In either case assign $x = (s, s+1)$ to be the follower of R_s and put $f_{s+1}(s) =$'waiting'.

This ends the stage and the construction.

Note that the sequence

$$(A_{s,0}, A_{s,1}, f_s, x_{q,s} \ : \ s \in \omega, \ q < s)$$

is recursive.

We put $A_i = \cup_{s\in\omega} A_{i,s}$. These are r.e. sets since $A_{i,s} \subseteq A_{i,s+1}$.

Verification

Claim. For each q

1. R_q acquires a permanent follower, i.e., there exist some stage s_0 such that for all $s > s_0$ the follower of R_q at stage s is that same as at stage s_0.
2. R_q is met, i.e, $\{e\}^{A_i} \neq A_{1-i}$
3. R_q acts at most finitely many times.

Proof

This is the main claim and it is proved by induction on q.

So suppose that (3) is true for all $q' < q$. Then there is a stage s_0 such that some $q' < q$ acted and no such $q' < q$ acts after stage s_0. Then the follower x_q of R_q appointed at stage s_0 is the permanent follower of R_q. Furthermore $f_{s_0}(q) =$'waiting'.

Suppose $q = 2e + i$. After stage s_0 there are two possibilities:
(a) for some $s > s_0$ we have that $\{e\}_s^{A_{i,s}}(x_q) \downarrow = 0$ with use less than s or
(b) not (a).

Suppose (a). In this case since no higher priority q' acts after stage s_0 then R_q will act. Hence x_q is put into A_{1-i}. Furthermore all other followers of lower priority requirements appointed now or at future stages will be larger than the use of the computation $\{e\}_s^{A_{i,s}}(x_q)$ (we assume that $s \leq \langle q', s\rangle$). Hence

$$\{e\}^{A_i}(x_q) \downarrow = 0 \neq 1 = A_{1-i}(x_q)$$

Suppose (b). In this case it must be that either

$$\{e\}^{A_i}(x_q)\uparrow \text{ or } \{e\}^{A_i}(x_q)\downarrow\neq 0.$$

In either case x_q is never put into A_{1-i} - this is because the possible followers of two distinct requirements are disjoint and no follower is used again for the same requirement. So $A_{1-i}(x_q) = 0 \neq \{e\}^{A_i}(x_q)$ and thus R_q is met.

So as we see R_q will act at most one more time after stage s_0 and so it acts only finitely many times. This proves the Claim and the Theorem.
QED

We say that R_q is injured when it is made to appoint new followers and start over. Hence, the terminology 'finite injury priority argument'.

Corollary 26.3 *There exists a set A which is r.e. and $0 <_T A <_T 0'$.*

Proof
Since 0 and $0'$ are $\leq_T$ comparable to every r.e. set it must be that both A_i from the Friedberg-Muchnik Theorem are strictly in between.
QED

Exercise 26.4. (Trachtenbrock)
Define A is auto-reducible iff there exists e such that for all x,

$$\{e\}^{A\setminus\{x\}}(x)\downarrow= A(x).$$

Prove
(a) For all B there exists $A \equiv_m B$ such that A is auto-reducible.
(b) There exist an r.e. A which is not auto-reducible.
(c) There exist a low r.e. A which is not auto-reducible.
(d)* There exist a r.e. $A \equiv_T K$ which is not auto-reducible?

27 Embedding in the r.e. degrees

We define

$$A_n = \{x \ : \ \langle n, x\rangle \in A\}$$

and

$$\oplus_{k\neq n} A_k = \{\langle k, x\rangle \in A \ : \ k < \omega \text{ and } k \neq n\}.$$

Theorem 27.1 *There exists an r.e. set A such that for every n*

$$A_n \not\leq_T \oplus_{k\neq n} A_k$$

Proof
This is a minor modification of the Friedberg-Muchnic argument (Theorem 26.2).

Our requirements are:
$R_{\langle e,n\rangle}$ $\{e\}^{\oplus_{k\neq n} A_k} \neq A_n$
for $e, n \in \omega$. And the construction is nearly the same:
At stage $s+1$ look for the least $q = \langle e, n\rangle < s$ such that

1. $f_s(q) =$'waiting' and
2. $\{e\}_s^{\oplus_{k\neq n} A_{k,s}}(x) \downarrow= 0$ with use less than s where $x = x_{q,s}$ is the follower of R_q.

If we find such a q then we take the following actions:

1. Put
$$A_{s+1} = A_s \cup \{\langle n, x\rangle\}$$
2. Set $f_{s+1}(q) =$'acted'.
3. Reappoint followers for lower priority requirements, i.e. for each $q' > q$ with $q' < s+1$ put $x = \langle q', s+1\rangle$ to be the follower of $R_{q'}$.
4. Restart lower priority requirements, for each $q' > q$ put
$$f_{s+1}(q') = \text{'waiting'}.$$

Finally, assign $x = (s, s+1)$ to be the follower of R_s and $f_{s+1}(s) =$'waiting'.

The verification is virtually the same as in the Friedberg-Muchnic Theorem.
QED

Corollary 27.2 *Every recursive partially ordered set embeds into the r.e. degrees $\mathcal{R}$.*

Proof
Let $\mathbb{P} = (\omega, \trianglelefteq)$ be a partial order with $\trianglelefteq$ a recursive binary relation on ω. Define $J(p) = \{\langle q, x\rangle \in A \ : \ q \trianglelefteq p\}$ and let $j(p) = deg(J(p))$. Then

$$j : \mathbb{P} \to \mathcal{R}$$

is an order preserving embedding.
QED

Exercise 27.3. Prove there exists a recursive partial order $\mathbb{P}_0 = (\omega, \leq_0)$ such that every countable partial order $\mathbb{P}_1$ can be embedded into it, i.e., there exists a 1-1 mapping $j : \mathbb{P}_1 \to \mathbb{P}_0$ such that $p \leq_1 q$ iff $j(p) \leq_0 j(q)$.

Hint: Construct $\mathbb{P}_0$ so that for every pair of finite posets $\mathbb{P}_1 \subseteq \mathbb{P}_2$ and embedding $j_1 : \mathbb{P}_1 \to \mathbb{P}_0$ there is an embedding $j_2 : \mathbb{P}_2 \to \mathbb{P}_0$ with $j_1 \subseteq j_2$.

It follows from this exercise that every countable partial order embeds into the r.e. degrees.

Exercise 27.4. Prove that for every creative set A there exist a set B which is r.e. and disjoint from A but cannot be separated from it by a recursive set. Prove that there exists disjoint r.e. sets A_0 and A_1 which are recursively inseparable but not creative. Hint: Construct A_0 and A_1 as in Theorem 26.2 with the additional requirements:

$$R_e \quad \psi_e = D \to D \text{ does not separate } A_0 \text{ and } A_1$$

28 Limit Lemma and Ramsey Theory

Lemma 28.1 *(The Limit Lemma) Suppose $g \in \omega^\omega$, then*
$g \leq_T 0'$
iff
there exists $f : \omega \times \omega \to \omega$ recursive such that for all n

$$\lim_{s\to\infty} f(n, s) = g(n).$$

Proof
Suppose $g = \{e\}^{0'}$. Let $(0'_s \ : \ s \in \omega)$ be a recursive enumeration of $0'$, e.g., $0'_s = \{e < s \ : \ \{e\}_s(e)\downarrow\}$. Define

$$f(n, s) = \begin{cases} 1 & \text{if } \{e\}_s^{0'_s}(n)\downarrow \\ 0 & \text{otherwise} \end{cases}$$

Then $g(n) = \lim_{s\to\infty} f(n,s)$.

For the converse, suppose that $g(n) = \lim_{s\to\infty} f(n,s)$ where f is recursive. For each n using an oracle for $0'$ we can compute s_0 so that for every $s > s_0$ we have that $f(n,s) = f(n,s_0)$.

(Try $s_0 = 0$ and ask the oracle if the computation that searches for a change in f ever terminates. If yes, try $s_0 = 1$, etc. Continue incrementing s_0 until the oracle says that beyond this stage f does not change.)

It follows that $g(n) = f(n,s_0)$. Hence there is an algorithm with oracle $0'$ which computes g.
QED

Definition 28.2 $[X]^n = \{s \subseteq X \;:\; |X| = n\}$

Ramsey Theorem says that for every $n, k < \omega$ and $f : [\omega]^n \to k$ there is $H \in [\omega]^\omega$ such that $f \restriction [H]^n$ is constant. This H is called homogeneous for f.

Example 28.3 *(Jockusch, Spector) There is a recursive $f : [\omega]^3 \to 2$ such that $0' \leq_T H$ for every infinite H which is homogeneous for f.*

Proof
Define

$$f(\{e_0 < s_1 < s_2\}) = \begin{cases} 0 & \text{if } \forall e < e_0 \;\; (\{e\}_{s_1}^{0'_{s_1}} \downarrow \text{ iff } \{e\}_{s_2}^{0'_{s_2}} \downarrow) \\ 1 & \text{otherwise.} \end{cases}$$

If H is an infinite homogeneous set for f, then f must map $[H]^3$ to 1 since every infinite set H contains a triple which f maps to 1.
QED

Example 28.4 *(Jockusch) There is a recursive $f : [\omega]^2 \to 2$ such that there does not exist an infinite recursive H which is homogeneous for f.*

Proof
Construct $b \in 2^\omega$ with the properties:

1. $b \leq_T 0'$ and
2. for every e if W_e is infinite then there are $n, m \in W_e$ such that $b(n) = 0$ and $b(m) = 1$.

By the limit Lemma there is a recursive $g : \omega^2 \to 2$ such that

$$b(n) = \lim_{s\to\infty} f(n,s).$$

Then f cannot have an infinite recursive homogeneous set H. For suppose $H = \{h_k : k < \omega\}$ is a strictly increasing computable enumeration of H. Then for $k < l$ we would have to have that $f(h_k, h_l) = b(h_k)$ and so b will be constant on H.
QED

See also Corollary 44.5 for another proof. Seetapun has shown that every recursive $f : [\omega]^2 \to 2$ has an infinite homogeneous set which does not compute $0'$.

29 A low simple set

Another way to prove that some r.e. degree is nontrivial is to construct a low simple set A. Since a simple set is not recursive we have that $0 <_T A$. Low means that $A' \equiv_T 0'$ so $A <_T 0'$ by Lemma 18.5.

Theorem 29.1 *There exists a low simple set A, i.e. $A' \equiv 0'$ and A is simple.*

Proof
We make the degree of A low by a strategy that is suggested by the proof of the limit lemma, namely we would like to use

$$f(e,s) = \begin{cases} 1 & \text{if } \{e\}_s^{A_s}(e)\downarrow \\ 0 & \text{otherwise} \end{cases}$$

to show that $A' \leq_T 0'$. That is, $A'(e) = \lim_{s\to\infty} f(e,s)$. If $e \in A'$ then it is easy to see that $f(e,s) = 1$ for all sufficiently large s. The problem then is to make sure that if $f(e,s) = 1$ for infinitely many s, then $e \in A'$.

So we make the following requirements:

N_e $(\exists^\infty s\ \{e\}_s^{A_s}(e)\downarrow) \to \{e\}^A(e)\downarrow$

In order to make sure that the set A is simple we have the following requirements:

P_e $(W_e \text{ infinite}) \to W_e \cap A \neq \emptyset$

The strategy for P_e is the same as for the Post Simple Set construction (Theorem 15.2), that is we wait for some $x \in W_{e,s}$ with $x > 2e$ and $A_s \cap W_{e,s} = \emptyset$ and put x into A_{s+1}.

The strategy for N_e is to wait until we see convergence and then try to prevent the computation from changing by restraining numbers less than the use of the computation from entering A.

The requirement P_e is positive since the strategy trys to put things into A while the requirement N_e is negative since it tries to keep things out of A.

Construction

At each stage in the construction we will have A_s and $r(e,s)$ for each e. We will always have that $r(e,s) = 0$ for $e \geq s$ so the function r is really a finite function.

Stage $s+1$. Look for the least $e < s$ such that

1. $W_{e,s} \cap A_s = \emptyset$
2. $\exists x > 2e$ with $x \in W_{e,s}$ and $x > r(e', s)$ for all $e' < e$.

For the least such e choose the least x as above and put $A_{s+1} = A_s \cup \{x\}$. We say in this case that P_e acted at stage $s+1$. If there is no such e put $A_{s+1} = A_s$.

Next we compute $r(e, s+1)$ for all $e < s+1$. If $\{e\}_s^{A_{s+1}}(e)\downarrow$, then put

$$r(e, s+1) = use(\{e\}_s^{A_{s+1}}(e)$$

otherwise put $r(e, s+1) = 0$.

This is the end of the construction. We let $A = \cup_{s\in\omega} A_s$ which is r.e.

Verification.

Claim.

1. P_e is met.
2. N_e is met.
3. $\lim_{s\to\infty} r(e,s) = r(e) < \infty$ exists.

Proof
We prove this by induction on e. Note that each P_e can act at most once, since after it acts W_e and A are no longer disjoint. Assume the claim is true for every $e' < e$.

(1) By induction we have some s_0 such that for all $s > s_0$ and $e' < e$ that $r(e', s) = r(e')$. Put

$$R = max\{r(e') \ : \ e' < e\}.$$

We can also choose s_0 so large that no $P_{e'}$ for $e' < e$ acts after stage s_0 since each $P_{e'}$ acts at most once. Suppose that W_e is infinite. It follows that at some stage $s > s_0$ there will be a $x \in W_{e,s}$ such that $x > 2e + R$. At stage $s + 1$ either $A_s \cap W_{e,s} \neq \emptyset$ or P_e will act. In either case P_e is met.

(2) Choose s_0 so that no $P_{e'}$ for $e' \leq e$ acts after stage s_0. This means that after stage s_0 no positive requirement can ever injure a computation of N_e. Hence if there is some $s_1 > s_0$ such that $\{e\}_{s_1}^{A_{s_1}}(e) \downarrow$ then no $x < use\{e\}_{s_1}^{A_{s_1}}(e)$ will ever enter A. It follows that this is the final computation and therefore $\{e\}^A(e) \downarrow$ with the same computation as at stage s.

(3) As above, either we never see convergence and then $r(e, s) = 0$ for all $s > s_0$ or we see convergence and then $r(e, s) = r(e, s_1)$ for all $s > s_1$.

This finishes the proof of the Claim and the Theorem.
QED

Exercise 29.2. (From Soare) A set A is auto-reducible iff there exists e such that for every x we have

$$\{e\}^{A \setminus \{x\}}(x) \downarrow = A(x).$$

Prove there is a r.e. set which is not auto-reducible. Extra credit: Prove that there exists a A low simple set which is not auto-reducible.

30 Friedberg splitting Theorem

Theorem 30.1 *(Friedberg Splitting) Every r.e. set which is not recursive is the disjoint union of two recursively enumerable sets which are not recursive.*

Proof

Suppose $B = \{b_s : s < \omega\}$ is a one-one recursive enumeration of the nonrecursive set B. We will decide at each stage to put b_s into either A_0 or A_1. Hence at any stage s we will have

$$B_s = \{b_t : t < s\} = A_0^s \sqcup A_1^s$$

where $\sqcup$ stands for disjoint union.

The requirements are:

R_{2e+i} $\quad W_e \neq \overline{A_i}$

The strategy is to try to make $A_i \cap W_e \neq \emptyset$.

Stage s

Find the least $2e + i < s$ (if any) such that

1. $W_e^s \cap A_i^s = \emptyset$ and
2. $b_s \in W_e^s$.

For the least such put b_s into A_i, i.e.,

$$A_i^{s+1} = A_i^s \cup \{b_s\} \text{ and } A_{1-i}^{s+1} = A_{1-i}^s.$$

In this case, we say that R_{2e+i} acted at stage s.

If there is no such $2e + i$ put b_s into A_0. This ends the construction.

Verification

Suppose for contradiction that $\overline{A_i} = W_e$. Since $A_i \subseteq B$ we know that

$$B \cup W_e = \omega.$$

We show that B is computable. Note that each requirement can act at most once. Choose a stage s_0 so that for any $q < 2e + i$ if R_q every acts it has already acted before stage s_0.

To compute B: Input x. Find any $s > s_0$ such that $x \in B_s \cup W_e^s$.

Case 1. $x \in B_s$. Hence $x \in B$.

Case 2. $x \in W_e^s \setminus B_s$. We claim that $x \notin B$. If it were then for some $t > s > s_0$ we would have $x = b_t$ and at that stage we would put b_t into A_i. But we are assuming $A_i \cap W_e = \emptyset$ and this would be a contradiction.
QED

Exercise 30.2. Suppose B is a r.e. set which is not recursive. Prove there exists a partial recursive function f with domain B such that for every $n < \omega$ the set $f^{-1}\{n\}$ is not computable.

Exercise 30.3. Prove or disprove. There exists A_n for $n < \omega$ pairwise disjoint r.e. sets which are not recursive such that

$$\omega = \sqcup_{n<\omega} A_n.$$

Exercise 30.4. Define f is proper iff f is a partial recursive function and both the domain and range of f are nonrecursive subsets of ω. Prove that for every proper f that there exists proper f_0 and f_1 with f the disjoint union of f_0 and f_1. (We are identifying the functions with their graph.)

Exercise 30.5. Show that if B is r.e. but not recursive, then there exists A_i r.e. such that $B = A_0 \sqcup A_1$ and A_0 and A_1 cannot be separated by a recursive set. Hint: If ψ_e is total, show that there must be infinitely many s such that $\psi_{e,s}(b_s) \downarrow$.

31 Sacks splitting Theorem

Theorem 31.1 *(Sacks) Suppose* $0 <_T C \leq_T 0'$ *and* A *is r.e. Then there exists r.e. sets* A_0 *and* A_1 *such that*

1. *A is the disjoint union of A_0 and A_1, i.e., $A = A_0 \sqcup A_1$,*
2. *$C \not\leq_T A_i$ for $i = 0, 1$, and*
3. *A_i is of low degree for $i = 0, 1$, i.e., $A_i' \equiv_T 0'$.*

Proof
By the limit lemma there exists a recursive function $g : \omega \times \omega \to 2$ such that for every n

$$C(n) = \lim_{s\to\infty} g(s, n).$$

To simplify notation let $C_s(n) = g(s, n)$.

Let $A = \{a_s : s \in \omega\}$ be a 1-1 recursive enumeration of A. If A is finite or even recursive the result is trivially true, so we don't have to worry about that case. We will achieve the splitting of A by simply putting a_s into exactly one of the two sets A_0 or A_1 at stage $s + 1$.

The Requirements

The lowness of the sets will be achieved by the same requirements as in the low simple set proof:

$N_{e,i}$ $\quad (\exists^\infty s \;\; \{e\}^{A_{i,s}}(e)\downarrow) \rightarrow \{e\}^{A_i}(e)\downarrow$

Our new requirements are for each $e \in \omega$ and $i = 0, 1$:

$R_{e,i}$ $\quad \{e\}^{A_i} \neq C$

which we will write $R_q = R_{e,i}$ where $q = 2e+i$. If we meet each of these, then $C \not\leq_T A_i$ for $i = 0, 1$. The strategy used for meeting $R_{e,i}$ is to preserve the length of agreement between $\{e\}^{A_i}$ and C. This seems contradictory, since we want them to be different. The reason it succeeds is because otherwise we will be able to compute C.

For each q we will have two variables l_q and u_q which are the length of agreement and the use of some computations. We will use u_q to restrain for both N_q and R_q.

The Construction

At stage $s = 0$ put $A_{i,s} = \emptyset$ and put $u_q = l_q = 0$.

Stage $s + 1$.

Begin by computing the length of agreement l_q and the usage u_q for each $q < s + 1$:

Suppose $q = 2e + i$.

(a) If $\{e\}_s^{A_{i,s}}(e))\downarrow$, then:

$$u_q := \max\{u_q, use(\{e\}_s^{A_{i,s}}(e))\}.$$

(b) Next we adjust the length of agreement. There are two cases:

(1) For all $x \leq l_q$

$$\{e\}_s^{A_{i,s}}(x)\downarrow = C_s(x).$$

In this case we bump up the usage and increment l_q:

$$\begin{aligned} u_q &:= \max\{\ u_q, use(\{e\}_s^{A_{i,s}}(x))\ :\ x \le l_q\ \} \\ l_q &:= l_q + 1 \end{aligned}$$

(2) Not case (1). In this case we do not change l_q and u_q.

Now we take action. Find the least $q < s+1$ (if any) such that $a_s < u_q$. If $q = 2e + i$, then put a_s into the opposite set, A_{1-i}, i.e.,

$$A_{1-i,s+1} = A_{1-i,s} \cup \{a_s\}.$$

(Hence we protect the computations in (b)(1) for q from being injured.)

If no such q exists, then put a_s into A_0. This ends the stage and the construction.

The Verification

Now we verify that the construction works. We use the notation l_q^s and u_q^s to refer to the values of these variables at stage s.

Claim. For each q

(1) R_q is met,

(2) $\lim_{s\to\infty} l_q^s = L_q < \infty$,

(3) $\lim_{s\to\infty} u_q^s = U_q < \infty$, and

(4) N_q is met.

Proof

In the case of (2) and (3) since our variables are nondecreasing this just means that at some stage they stop growing. The Claim is proved by induction on q. So suppose it is true for all $p < q$ and let $R_q = R_{e,i}$

Proof of (1)

For contradiction assume that R_q is not met, i.e.,

$$\{e\}^{A_i} = C.$$

Subclaim (a). $\lim_{s\to\infty} l_q^s = \infty$.

To see why this is true, note that for any x there will be some stage s_0 where $C_s \restriction x = C \restriction x$ for all $s > s_0$ and also $\{e\}^{A_i} \restriction x$ will be same computations as $\{e\}_{s_0}^{A_{i,s_0}} \restriction x$, i.e., the use of the oracle has settled down. After s_0 the variable l_q will be incremented until it is at least x, if it isn't already. This proves subclaim (a).

Now go to a stage s_0 such that for all $s > s_0$

1. for all $p < q \quad u_p^s = U_p$ and
2. $a_s > \max\{U_p \ : \ p < q\}$.

Subclaim (b). If $s > s_0$ is a stage where l_q is incremented then

$$C(x) = \{e\}_s^{A_{i,s}}(x).$$

for any $x < l_q$

To see why this is true, note that u_q protects the computation $\{e\}_s^{A_{i,s}}(x)$ from ever changing since a_s is never beneath u_p for any higher priority $p < q$. This means that

$$\{e\}_s^{A_{i,s}}(x) = \{e\}^{A_i}(x).$$

But we are assuming $\{e\}^{A_i} = C$. This proves subclaim (b).

Now we get a contradiction to our assumption that C is not recursive. To compute $C(x)$ search for a stage $s > s_0$ where $l_q > x$ and it has just been incremented. Then $C(x) = \{e\}_s^{A_{i,s}}(x)$.

This contradiction proves the main Claim part (1) that R_q is met.

Proof of (2)

Since R_q is met there exists x such that either

(a) $\{e\}^{A_i}(x) \uparrow$ or

(b) $\{e\}^{A_i}(x) \downarrow \neq C(x)$.

Fix any such x. Go to a stage s_0 such that for all $s > s_0$

1. for all $p < q \quad u_p^s = U_p$,
2. $a_s > \max\{U_p \;:\; p < q\}$, and
3. $C_s(x) = C(x)$.

It is impossible that at some stage $s > s_0$ where $l_q > x$ that l_q is incremented. This is because at such a stage s

$$\{e\}_s^{A_{i,s}}(x) \downarrow = C_s(x).$$

For the rest of the construction u_q will protect the computation $\{e\}_s^{A_{i,s}}(x)$. But then

$$\{e\}^{A_i}(x) = \{e\}_s^{A_{i,s}}(x) = C_s(x) = C(x)$$

which contradicts the choice of x.

Proof of (3)

Note that u_q changes only when either l_q is incremented or when we see $\{e\}_s^{A_{i,s}}(e)$ converges. Hence if we go to a stage s_0 such that for all $s > s_0$

1. for all $p < q$ $u_p^s = U_p$,
2. $a_s > \max\{U_p \;:\; p < q\}$, and
3. $l_q^s = L_q$

then u_q will change at most once more, after which it protects the computation $\{e\}_s^{A_{i,s}}(e)$ from changing and never changes again.

Proof of (4)

The proof that N_q is met is the same as in the low simple set argument.

This ends the proof of the Claim and of the Sacks Splitting Theorem. QED

Proposition 31.2 *Suppose $A = A_0 \sqcup A_1$ is a disjoint union of r.e. sets A_0 and A_1, then $A \equiv_T A_0 \oplus A_1$.*

Proof
Clearly $A = A_0 \cup A_1 \leq_T A_0 \oplus A_1$. To see that $A_i \leq_T A$, input x and first ask the oracle if $x \in A$. If yes, enumerate A_0 and A_1 until x shows up.
QED

Corollary 31.3 *(Friedberg Splitting) Every r.e. set which is not recursive is the disjoint union of two r.e. sets which are not recursive.*

Proof
Take $C = A$. Then $A_i \not\geq_T A$ but if either is recursive then by Proposition 31.2 we get a contradiction.
QED

Corollary 31.4 *For every $c \in \mathcal{D}$ if $o < c < o'$, then there exists $a \in \mathcal{R}$ with $a|c$.*

Proof
Let $A = 0' = K$. By Proposition 31.2, $A = A_0 \oplus A_1$ where $C \not\geq_T A_i$ for both $i = 0, 1$. But then at most one of the A_i can be $\leq_T C$, since otherwise

$$0' \equiv_T A_0 \oplus A_1 \leq_T C.$$

QED

Corollary 31.5 *There exists $a_0, a_1 \in \mathcal{R}$ such that*

$$(a_0 \vee a_1)' \neq a_0' \vee a_1'$$

Proof
By the Theorem there exists low r.e. sets A_i such that $A_0 \oplus A_1 \equiv_T 0'$. Hence

$$a_0' \vee a_1' = o' < o'' = (a_0 \vee a_1)'$$

QED

Corollary 31.6 *No r.e. degree is minimal, in fact, beneath any nontrivial r.e. degree is a nontrivial low r.e. degree.*

Proof
Given r.e. set A which is not recursive, let $C = A$ and then we have low r.e. sets A_0 and A_1 which split A and $A \not\leq_T A_i$. Then for each i we have that $0 <_T A_i <_T A$.
QED

Exercise 31.7. (Welch) Prove there are low r.e. degrees a_0 and a_1 such that for every r.e. degree b there are r.e. degrees $b_0 \leq a_0$ and $b_1 \leq a_1$ with $b = b_0 \vee b_1$. Hint: Sacks split W.

32 Lachlan and Yates: minimal pair

Theorem 32.1 *(Lachlan, Yates) There exists a minimal pair of r.e. degrees, i.e.* $a_0, a_1 \in \mathcal{R} \setminus \{o\}$ *such that the only degree* b *with* $b \leq a_0$ *and* $b \leq a_1$ *is* $b = o$*.*

Proof

Requirements:

$P_{e,i}$ $\quad \psi_e \neq A_i$

N_{e_0,e_1} $\quad (\{e_0\}^{A_0} = \{e_1\}^{A_1} = B) \rightarrow B$ recursive.

Strategies:

For $P_{e,i}$ wait for $\psi_{e,s}(x) \downarrow= 0$ for some follower x and then put x into A_i.

For N_{e_0,e_1} restrain agreement to get (a) or (b):
(a) for some $l < \omega$ we have that $\{e_0\}^{A_0} \restriction l \downarrow= \{e_1\}^{A_1} \restriction l \downarrow$ and either
$(\{e_0\}^{A_0}(l) \uparrow$ or $\{e_1\}^{A_1}(l) \uparrow)$ or $(\{e_0\}^{A_0}(l) \downarrow\neq \{e_1\}^{A_1}(l) \downarrow)$
(b) $\{e_0\}^{A_0} = \{e_1\}^{A_1} = B$ and B is recursive by virtue of our restraining certain computations, that is, we can compute B by finding stages where we can be sure the approximate computation at that stage is the final one.

Outcomes:

For $P_{e,i}$ the outcomes are either to wait forever or to act at some time. We order them by { act < wait }.

For N_{e_0,e_1} the outcomes are either $l < \omega$ where l is the largest length of agreement which we see at a true stage or $\{\infty\}$ if the length of agreement has infinite limit. We use the ordering

$$\infty < \cdots < l+1 < l < \cdots < 2 < 1 < 0$$

because it is traditional to take limit infimums (rather than limsups) in the outcome tree to determine the truth path.

The outcomes are $\Lambda = \{$act,wait $\} \cup \{\infty\} \cup \omega$. The tree of outcomes is $\Lambda^{<\omega}$. At each stage s in the construction we will have recursively constructed $f_s \in \Lambda^s$ which is an approximation to the true path, i.e., the eventually correct outcomes.

If $\alpha \in \Lambda^n$ where $n = 2\langle e_0, e_1\rangle$ then α works on the requirement N_{e_0,e_1}. If $\beta \in \Lambda^n$ where $n = 2m+1$ and $m = 2e+i$, then β works on the requirement $P_{e,i}$.

Supplementary variables:

For each such β working on a positive requirement we have a restraint variable $R_\beta \in \omega$. Also for each such β we let

$$F_\beta = \{\langle \beta, x\rangle : x \in \omega\}$$

be the followers of β. These could be any pairwise disjoint family of uniformly recursive infinite subsets of ω.

For each α working on a negative requirement we have two variables l_α and u_α (length of agreement and the usage of some computations).

The Construction:

Stage $s = 0$. Put $A_{0,0} = A_{1,0} = \emptyset$ and $f_0 = \langle\rangle$, and put all supplementary variables, $R_\beta, l_\alpha, u_\alpha$ equal to zero.

Stage $s+1$. Given $A_{0,s}, A_{1,s}$, and $f_s \in \Lambda^s$ proceed as follows.

Action:

Look for the least $\beta \subsetneq f_s$ working on a positive requirement $P_{e,i}$ such that

(1) $f_s(|\beta|) =$'wait' and

(2) there exist $x > R_\beta$ with $x \in F_\beta$ and $x < s$ such that $\psi_{e,s}(x) \downarrow= 0$.
Put the least such x into A_i, i.e.,

$$A_{i,s+1} = A_{i,s} \cup \{x\}.$$

In this case we say that β and $P_{e,i}$ acted at stage $s + 1$. If no such β exists, then no action is taken.

Update variables:

Define $f_{s+1}\restriction n$ for $n \leq s+1$ by induction on n. At the same time we may update the supplementary variables for each $\gamma \subseteq f_{s+1}$.

Case $\beta = f_{s+1}\restriction n$ where β is working on $P_{\hat{e},\hat{i}}$.

If $P_{\hat{e},\hat{i}}$ has acted at some stage $\leq s+1$ then put $f_{s+1}(n) =$'act'. Otherwise $f_{s+1}(n) =$'wait'.

Define R_β to be the maximum of the following sets:
(1)$\{u_\alpha : \alpha <_{lex} \beta\}$ where $\alpha <_{lex} \beta$ means that there exists $k < \min(|\alpha|, |\beta|)$ such that $\alpha\restriction k = \beta\restriction k$ and $\alpha(k) < \beta(k)$ in the ordering of outcomes.
(2) $\{u_\alpha : \alpha \subsetneq \beta \text{ and } \beta(|\alpha|) \neq \infty\}$.

Remarks. β preserves computations of α's which are lexicographically to its left because α's want β's to their right to respect their computations. β also respects computations directly below it except for those which β thinks will have an infinite length of agreement.

Case $\alpha = f_{s+1}\restriction n$ and α is working on N_{e_0,e_1}.

We begin by asking:

Does $\{e_0\}_{s+1}^{A_0,s+1}(x) \downarrow= \{e_1\}_{s+1}^{A_1,s+1}(x) \downarrow$ for every $x \leq l_\alpha$?
If yes, we put $f_{s+1}(n) = \infty$ and we set:

$$\begin{aligned} u_\alpha &:= \max\{u_\alpha, use(\{e_i\}_{s+1}^{A_i,s+1}(x)) : x \leq l_\alpha, i = 0, 1\} \\ l_\alpha &:= l_\alpha + 1 \end{aligned}$$

If no, we put $f_{s+1}(n) = l_\alpha$ and make no changes in the variables.

Remarks. If we see expansion in the length of agreement over what it was when last we set it, we guess optimistically that the length of agreement will expand forever. If we don't see this expansion, we pessimistically guess we will never see another expansion. (At least on the stages which go thru α.)

Verification.

We begin by defining the true path $f \in \Lambda^\omega$. We define $f \restriction n$ by induction on n. First let

$$T_n = \{s > n : f \restriction n \subseteq f_s\}$$

these are the true stages and note that $T_n \subseteq T_{n-1}$. The set T_n is a recursive set which (by induction) is infinite. Define $f(n)$ by

$$f(n) = \liminf_{s \in T_n} f_s(n).$$

If $\beta = f \restriction n$ is working on $P_{e,i}$, then $f(n) =$'act' if $P_{e,i}$ every acts, and otherwise $f(n) =$'wait', meaning we wait forever. In the case $\alpha = f \restriction n$ is working on a negative requirement $f(n)$ will be ∞ if there are infinitely many $s \in T_n$ in which the length of agreement l_α has been incremented and otherwise it will be the final value of l_α.

Claim. For each n the requirement that $f \restriction n$ is working on is met.
Proof

Case $f \restriction n = \beta$ is working on $P_{e,i}$.

If $f(n) =$'act', then for some x we put x into A_i at a stage s where we saw $\psi_{e,s}(x) \downarrow = 0$. But then $A_i(x) = 1 \neq \psi_e(x)$.

If $f(n) =$'wait', let us first prove that R_β does not change at any stage $s \geq \min(T_n)$. We first note that for every $s > \min(T_n)$ that it is not true that $f_s <_{lex} \beta$. Why? Suppose $f_s \restriction k = \beta \restriction k$ and $f_s(k) < \beta(k)$. If $\beta(k) =$'wait' and $f_s(k)$='act', then we get a contradiction, since then β is not on the true path f. In the case of a negative requirement $\alpha = \beta \restriction k$ then $\beta(k) = l < \omega$ (since nothing is to the left of ∞), but this would mean that the true path would go to the left of β. It follows that for every $s \in T_n$ the variables $\{u_\alpha : \alpha <_{lex} \beta\}$ will be what they were at the stage $s = min(T_n)$. Similarly for any u_α with $\alpha \subseteq \beta$ and $\beta(|\alpha|) \neq \infty$ these variables will have also reached their maximum since u_α is only changed when l_α is incremented.

To see that $P_{e,i}$ is met in this case let R^*_β be this final value of R_β. Let $x \in F_\beta$ with $x > R^*_\beta$. It is not the case that $\psi_e(x) \downarrow = 0$, because if this ever happened then for some large enough stage $s \in T_n$ the worker β would have acted (either putting this or some smaller x into A_i. Since x is never put into A_i the requirement is met because $\psi_e(x) \neq 0 = A_i(x)$.

Case $f\upharpoonright n = \alpha$ is working on N_{e_0,e_1}.

Subcase $f(n) = l$

Then for every $s \in T_{n+1}$ the length of agreement was less than $l+1$, i.e. for some $x \leq l+1$ it was not true that:

$$\{e_0\}_s^{A_0,s}(x)\downarrow = \{e_1\}_s^{A_1,s}(x)\downarrow$$

otherwise we would have incremented l_α. It follows that

$$\neg(\{e_0\}^{A_0} = \{e_1\}^{A_1} = B)$$

and so N_{e_0,e_1} is satisfied.

Subcase $f(n) = \infty$

Then we claim that B is recursive. To see this suppose $s_1 < s_2$ are successive stages in T_{n+1}. Note that $\alpha = f_{s_1}\upharpoonright n = f_{s_2}\upharpoonright n$ and $f_{s_1}(n) = f_{s_2}(n) = \infty$. See Figure 5.

This means that l_α was incremented at each stage s_i, say $l-1$ to l at stage s_1 and l to $l+1$ at stage s_2. At stage s_1 before any action the two computations agreed:

$$\{e_0\}_{s_1}^{A_0,s_1}\upharpoonright l\downarrow = \{e_1\}_{s_1}^{A_1,s_1}\upharpoonright l\downarrow .$$

If $\beta_1 \subseteq f_{s_1}$ is the node which acted at stage s_1 (if any), then it must be that $\alpha \subseteq \beta_1$ and $\beta_1(n) = \infty$. This action could destroy either the left side or ride side of this agreement but not both, since some x may be put into A_0 or A_1 but not both. The variable u_α is set to protect the surviving side in subsequent stages. At stages s with $s_1 < s < s_2$ any acting node β must either be above $\alpha\hat{}\langle l_\alpha\rangle$ or be lexicographically to the right of α as β' is in the figure. But this means that $R_\beta \geq u_\alpha$ and so the action at stage s cannot damage the surviving side. At stage s_2 we increment l to $l+1$ which means that the destroyed side must have come back and equaled the surviving side.

This means that for each $s, s' \in T_{n+1}$ with $s < s'$ and $x < l_\alpha^s$:

$$\{e_0\}_s^{A_0,s}(x)\downarrow = \{e_0\}_{s'}^{A_0,s'}(x)\downarrow .$$

The two computations may be different but they output the same value (and the same for e_1). Hence, assuming $\{e_0\}^{A_0} = B$, to compute $B(x)$ search for a stage $s \in T_{n+1}$ such that $x < l_\alpha^s$ and then $B(x) = \{e_0\}_s^{A_0,s}(x)$. It follows that B is recursive. This proves the Claim and the minimal pair theorem.

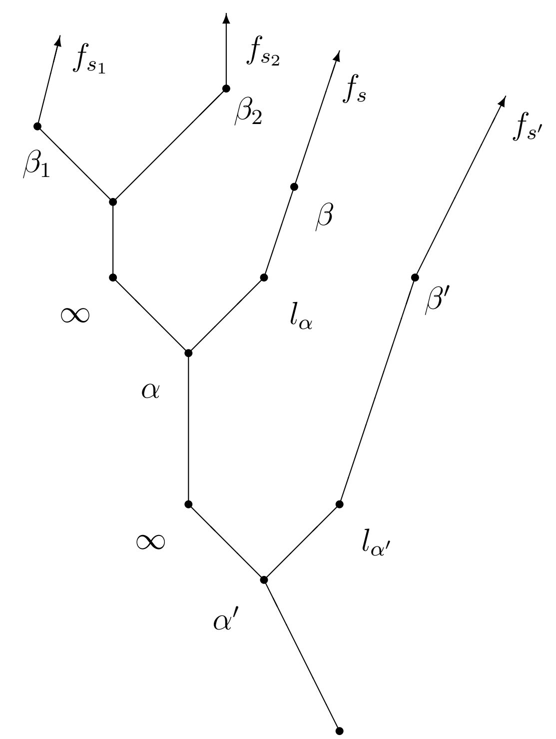


Figure 5: $s_1 < s, s' < s_2$

QED

Exercise 32.2. Put the Friedberg-Muchnik argument on a tree of outcomes. Show there is no injury on the true path.

Exercise 32.3. Put the low simple non-auto-reducible set construction on a tree of outcomes. Prove the construction works. Show that there is no injury on the true path.

Exercise 32.4. (From Cooper) Show that there is minimal pair A_0 and A_1 such that $(A_0 \oplus A_1)' \equiv_T 0'$.

33 Friedberg: A one-one enumeration of the r.e. sets

Theorem 33.1 *(Friedberg, Enumeration without repetition) There exists an r.e. set U such that*

1. $\{U_e \; : \; e \in \omega\}$ *is the set of all r.e. sets and*
2. $U_{e_1} \neq U_{e_2}$ *for all* $e_1 \neq e_2$

Proof

We will first construct an r.e. set V and then modify it to get U. The requirements are:

$R_e \quad \forall \hat{e} < e \;\; (W_{\hat{e}} \neq W_e) \to W_e = V_x$ for some unique x.

The strategy for meeting this requirement is to appoint a follower x. As long as it looks like $\forall \hat{e} < e \;\; (W_{\hat{e}}\restriction x \neq W_e\restriction x)$ keep enumerating W_e into V_x. Otherwise make it a disloyal follower and put it into the garbage. What do we do with V_x when x is a disloyal follower? We make it into an initial segment.

Definition 33.2 *$A \subseteq \omega$ is an initial segment iff $A = \emptyset$ or $A = \omega$ or there exists $n < \omega$ such that $A = [0, n] =^{def} \{i < \omega \; : \; 0 \leq i \leq n\}$.*

So our modified requirement is:

R_e If $\forall \hat{e} < e \ \ (W_{\hat{e}} \neq W_e)$ and W_e is not an initial segment, then $W_e = V_x$ for some unique x.

At stage $s+1$ in our construction we have the following sets:

1. F_s the followers
2. a 1-1 mapping from F_s to ω which tells us that x is the follower of e, say $f_s(x) = e$
3. D_s the disloyal former followers
4. $(V_{x,s} \ : \ x \in F_s \cup D_s)$
5. a nondecreasing variable g_s keeping track of last initial segment assigned to a disloyal follower.

The sets F_s and D_s will be disjoint finite sets whose union is an initial segment.

Construction

Stage $s+1$

Let $s = \langle e, ?\rangle$. (So we visit each e infinitely often.)

If no follower is assigned to R_e, let $x = min(\overline{F_s \cup D_s})$ and assign x to be the follower of R_e. Put $F_{s+1} = F_s \cup \{x\}$ and end the stage.

If x is the follower of R_e and

1. $\forall \hat{e} < e$
$$W_{\hat{e},s+1} \cap [0.x] \neq (W_{e,s+1}) \cap [0,x]$$
2. $W_{e,s+1} \cap [0,x]$ is not an initial segment

then put $V_{x,s+1} = V_{x,s} \cup W_{e,s+1}$ and end the stage. Actually in this case $V_{x,s} \subseteq W_{e,s}$ so we could have said put $V_{x,s+1} = W_{e,s+1}$.

If x is the follower of R_e and either of those two conditions fails then

1. change x into a disloyal follower, i.e., $F_{s+1} = F_s \setminus \{x\}$ and $D_{s+1} = D_s \cup \{x\}$,

2. let g_{s+1} be the minimum $g > g_s$ such that $V_{e,s} \subseteq [0, g]$, and
3. permanently assign V_x to be $[0, g_{s+1}]$, i.e., set $V_{x,s+1} = [0, g_{s+1}]$ and never change V_x again.

End the stage.

Verification

Claim 1. The following are equivalent for any e:

1. W_e is not an initial segment of ω and $W_e \neq W_{\hat{e}}$ for each $\hat{e} < e$.
2. R_e obtains a permanent follower x and hence $V_x = W_e$.

Proof
Suppose condition 2 holds. Then R_e obtains a permanent follower x. Then for all stages $s+1$ after x is appointed and for which $s = \langle e, ? \rangle$, we have that $W_{e,s} \cap [0, x]$ is not an initial segment and $W_{e,s} \cap [0, x] \neq W_{\hat{e},s} \cap [0, x]$ for each $\hat{e} < e$. Condition (1) follows since there are infinitely many such stages.

Suppose that condition 1 holds. Choose y so that $W_e \cap [0, y]$ is not an initial segment and

$$W_e \cap [0, y] \neq W_{\hat{e}} \cap [0, y]$$

for every $\hat{e} < e$. Go to some stage s_0 where

$$W_{e,s_0} \cap [0, y] = W_e \cap [0, y]$$

and

$$W_{\hat{e},s_0} \cap [0, y] = W_{\hat{e}} \cap [0, y]$$

for every $\hat{e} < e$. If R_e has no permanent follower then infinitely many followers are appointed to it. Hence some follower $x > y$ will be appointed after stage s_0. But such a follower will always remain loyal.
QED

Let $D = \cup_{s \in \omega} D_s$ be the set of disloyal followers. Then $\overline{D}$ is the set of permanent followers.

Claim 2.

1. $\{V_x : x \in \overline{D}\}$ is the set of r.e. sets which are not initial segments.
2. There exist a recursive set G such that
$$\{[0,n] : n \in G\} = \{V_x : x \in D\}.$$
3. $V_x \neq V_{x'}$ unless $x = x'$.

Proof

Part (1) follows from Claim 1.

For Part (2), since the sequence g_s is non-decreasing we see that

$$G = \{g_s : s \in \omega\}$$

is recursive.

For Part (3) note that there are two types of V_x. If x is a permanent follower of some R_e and then $V_x = W_e$ where W_e is not an initial segment and W_e is distinct from each $W_{\hat{e}}$. Or x is a disloyal follower at some stage $s+1$ and then $V_x = [0, g_{s+1}]$. Since the sequence g_s is bumped up each time it is used we see that the V_x for disloyal followers are distinct finite initial segments. This proves Claim.

QED

Let us show how to modify V to U to prove Friedberg's enumeration without repetition theorem. Note that V uniquely enumerates every r.e. set except ω, $\emptyset$, and the finite initial segments of the form $[0,n]$ where $n \notin G$. Let $\{x_n : 1 < n < \omega\}$ be a 1-1 recursive enumeration of $\overline{G}$. Now define U by $U_0 = \omega$, $U_2 = \emptyset$, $U_{2n} = [0, x_n]$ for $n > 1$, and $U_{2n+1} = V_n$.

QED

Definition 33.3 *A family of subsets $\mathcal{V}$ of ω is called an r.e. class iff there exists an r.e. set V such that*

$$\mathcal{V} = \{V_e : e \in \omega\}$$

where $V_e = \{x : \langle e, x\rangle \in V\}$. V is call an enumeration of $\mathcal{V}$. If $V_e \neq V_{e'}$ whenever $e \neq e'$ then V is called a Friedberg enumeration of $\mathcal{V}$.

Theorem 33.4 *If $\mathcal{V}$ is an r.e. class containing all initial segments, then $\mathcal{V}$ has a Friedberg enumeration.*

Proof
This is an obvious modification of the proof of Theorem 33.1.
QED

Example 33.5 *(Pour-El, Putnam) There is an r.e. class consisting of infinitely many one and two element sets which has no Friedberg enumeration.*

Proof
Take A to be any set which is r.e. but not recursive. Let $F_n = \{2n, 2n+1\}$ and $G_n = \{2n\}$. Then

$$\mathcal{V} = \{F_n \;:\; n \in \omega\} \cup \{G_n \;:\; n \in \overline{A}\}$$

is an r.e. class. This is because we just enumerate $2n+1$ into G_n turning it into F_n when n is enumerated into A). But $\mathcal{V}$ cannot have a Friedberg enumeration V since then:

$$\forall n \;\; (n \in \overline{A} \text{ iff } \exists x, y \;\; x \neq y \text{ and } 2n \in (V_x \cap V_y)).$$

QED

Example 33.6 *(Pour-El, Putnam) There is an infinite r.e. class $\mathcal{V}$ containing ω such that any enumeration of $\mathcal{V}$ must list ω infinitely many times.*

Proof
Let A be any r.e. set which is not recursive. Let $\mathcal{V}$ be the class of r.e. sets B such that $B \subseteq \overline{A}$ or $B = \omega$. To see that $\mathcal{V}$ is an r.e. class just enumerate each W_e into V_e as long as you see that $A_s \cap W_{e,s} = \emptyset$. If this ever fails, enumerate all of ω into V_e.

If there is an enumeration of $\mathcal{V}$ which only lists ω finitely many times, then there is an enumeration U of the elements of $\mathcal{V}$ which are not ω. But then

$$\overline{A} = \bigcup \{U_e \;:\; e \in \omega\}$$

would the be an r.e. set.
QED

It seems to require a more complicated proof than that for Theorem 40.6 to show:

Theorem 33.7 *(Friedberg) The class of graphs of partial recursive function has a Friedberg enumeration.*

Exercise 33.8. Prove that the family of recursive sets is an r.e. class and has a Friedberg enumeration.

Exercise 33.9. Prove that the family of r.e. sets which are not simple is an r.e. class and has a Friedberg enumeration.

34 Hypersimple sets

Definition 34.1 *Coding finite sets. For* $D \subseteq \omega$ *let* $x = \sum_{n \in D} 2^n$. *Write* $D_x = D$.

Definition 34.2 $(D_x : x \in R)$ *is a strong array iff* R *is an infinite recursive set and for every* $x, y \in R$ *we have* $D_x \cap D_y = \emptyset$ *whenever* $x \neq y$.

Definition 34.3 *A set* $A \subseteq \omega$ *is hypersimple iff* A *is r.e. ,* $\overline{A}$ *is infinite, and for every strong array* $(D_x : x \in R)$ *there exists* $x \in R$ *such that* $D_x \subseteq A$.

Proposition 34.4 *(Post)*
(1) Hypersimple implies simple.
(2) There is a simple set which is not hypersimple.
(3) There is a hypersimple set.

Proof
(1) If A is not simple, then there exists an infinite recursive set $R \subseteq \overline{A}$. Then $\{D_{2^x} : x \in R\}$ witnesses that A is not hypersimple.
(2) In Post's original construction of a simple set A (see Theorem 15.2) we constructed a simple set A by waiting until there was some $x \in W_{e,s}$ with $x > 2e$ and $W_{e,s} \cap A_s = \emptyset$ and then putting x into A. The reason that $\overline{A}$ was infinite was because for every e we had that $|[0, 2e] \cap A| \leq e$. This means that for every a we have that

$$[a, 4a] \cap \overline{A} \neq \emptyset$$

because $[a, 4a]$ is 3/4 of the interval $[0, 4a]$. So define $a_0 = 5$ and $a_{n+1} = 4a_n + 1$. Take x_n so that $D_{x_n} = [a_n, 4a_n]$ and note that $D_{x_n} \cap \overline{A} \neq \emptyset$ for each n so the recursive set $R = \{x_n : n < \omega\}$ witnesses that A is not hypersimple.
(3) This is a consequence of the following proposition, although originally Post gave a construction similar to his construction of a simple set.
QED

Proposition 34.5 *(Dekker) Deficiency sets are hypersimple.*

Proof
See Theorem 17.1. Suppose that $A = \{a_s : s \in \omega\}$ is a 1-1 recursive enumeration of A and A is not recursive. Define

$$D = \{s : \exists t > s \quad a_t < a_s\}.$$

As we saw before $A \equiv_T D$ and D is simple. A similar proof will show that D is hypersimple.

Suppose for contradiction that there exists a strong array $(D_x : x \in R)$ such that $D_x \cap \overline{D} \neq \emptyset$ for every $x \in R$.

Now we get a contradiction by showing that A is computable.

Input u. Find an $x \in R$ such that

$$u < \min\{a_s : s \in D_x\}.$$

Such an x exists, since a_s is a 1-1 enumeration and the D_x are pairwise disjoint. But now at least one of $t \in D_x$ is not deficient, so for all $s > t$ we have $a_s > a_t$. Hence $u \in A$ iff $u = a_s$ for some $s \leq \max D_x$.
QED

Exercise 34.6. Define A to be bdd-hypersimple iff A is r.e. , $\overline{A}$ is infinite, and for every strong array $(D_x : x \in R)$ such that there exists $N < \omega$ such that $|D_x| \leq N$ for all $x \in R$, there exists $x \in R$ such that $D_x \subseteq A$. Prove that bdd-hypersimple is equivalent to simple.

Definition 34.7 *For any set $A \subseteq \omega$ such that $\overline{A}$ is infinite define $\overline{a}_n$ to be the $(n+1)^{th}$ element of $\overline{A}$, i.e.,*

$$\overline{A} = \{\overline{a}_0 < \overline{a}_1 < \cdots < \overline{a}_n < \cdots\}.$$

Proposition 34.8 *For any r.e. set A with $\overline{A}$ infinite the following are equivalent:*

1. *A is hypersimple.*

2. *For any recursive increasing sequence $n_k < n_{k+1}$ there are infinitely many k with $[n_k, n_{k+1}) \subseteq A$.*

3. For any recursive $f \in \omega^\omega$ there are infinitely many k such that $f(k) < \overline{a}_k$.

Proof

$(1) \to (2)$. This is clear since if $D_{x_k} = [n_k, n_{k+1})$, then $R = \{x_k \;:\; k < \omega\}$ is a strong array. There are infinitely many since $R(l) =^{def} \{x_k : k > l\}$ is a strong array for any l.

$(2) \to (3)$. Given a recursive f construct a recursive sequence $n_{k+1} > n_k$ with the property that $f(n_k + 1) < n_{k+1}$ for each k. For any k such that $[n_k, n_{k+1}) \subseteq A$ note that $\overline{A} \cap [0, n_{k+1}) \subseteq [0, n_k)$ and so $\overline{a}_{n_k+1} = (n_k + 1)^{th}$ element of $\overline{A}$ must be greater than n_{k+1}. Hence $f(n_k + 1) < \overline{a}_{n_k+1}$.

$(3) \to (1)$. Suppose A is not hypersimple and hence there exists a strong array $(D_x \;:\; x \in R)$ such that $D_x \cap \overline{A} \neq \emptyset$ for all $x \in A$. Let $\{x_n \;:\; n \in \omega)$ be a 1-1 recursive enumeration of R and define

$$f(n) = 1 + \max(\cup_{m \leq n} D_{x_m})$$

Then $|\overline{A} \cap [0, f(n))| > n$ and so $f(n) > \overline{a}_n$.
QED

Exercise 34.9 *Suppose A is hypersimple and $f : \omega \to \omega$ is recursive. Prove there exist an infinite recursive set C such that $f(n) < \overline{a}_n$ for all $n \in C$.*

Exercise 34.10. Prove that for every r.e. set $A \subseteq \omega$ if $\overline{A}$ is infinite, then there exists a hypersimple set $B \supseteq A$.

Consider propositional logic with the set of atomic letters

$$\{P_n \;:\; n \in \omega\}.$$

For any propositional sentence ψ and subset $A \subseteq \omega$ define

$$A \models \psi$$

inductively by

$$A \models P_n \text{ iff } n \in A$$

$$A \models \neg\psi \text{ iff not } A \models \psi$$

$$A \models (\psi \vee \theta) \text{ iff } (A \models \psi \text{ or } A \models \theta)$$

and so forth for the other logical symbols.

By coding symbols as elements of ω and thinking of sentences as strings of symbols or finite sequences of elements of ω, we identify the set of propositional sentences with a recursive subset of ω, SENT. The details of this coding are left to the reader.

The following notion is known as truth-table (tt) reducibility.

Definition 34.11 *$A \leq_{tt} B$ iff there exists a recursive sequence*

$$(\theta_n \in SENT \ : \ n \in \omega)$$

such that for all $n \in \omega$

$$n \in A \text{ iff } B \models \theta_n$$

Note: It is easy to see that $A \leq_{tt} C$ and $B \leq_{tt} C$ implies $(A \cap B) \leq_{tt} C$ and $\overline{A} \leq_{tt} C$. Hence the family of sets which are truth-table reducible to C is closed under finite boolean combinations. It is easy to see that $\leq_m$-reducible is stronger than $\leq_{tt}$, and $\leq_{tt}$ is stronger than $\leq_T$.

Proposition 34.12 *(Nerode) The following are equivalent:*

1. *$A \leq_{tt} B$.*
2. *There exist e with the property that*

   $$\forall X \ \forall x \ \{e\}^X(x) \downarrow$$

   and $\{e\}^B = A$.
3. *There exists e and $f \in \omega^\omega$ recursive such that*

   $$\forall x \ \{e\}^B_{f(x)}(x) \downarrow$$

   and $\{e\}^B = A$.

Proof

$(1) \to (2)$. Given $(\theta_n \ : \ n \in \omega)$ witnessing that $A \leq_{tt} B$, it is easy to construct an oracle machine e such that for any input x and oracle X that $\{e\}^X(x) \downarrow = 1$, if $X \models \theta_x$ and $\{e\}^X(x) \downarrow = 0$, if $X \models \neg\theta_x$.

$(2) \to (3)$. We show that the same e works. Input x and let

$$T_x = \{\sigma \in 2^{<\omega} \ : \ \{e\}^\sigma_{|\sigma|}(x) \uparrow\}.$$

The trees T_x are uniformly recursive in x. By Konig's tree lemma, since T_x has no infinite branch, it is finite. Therefore we can compute the least n such that for all $\sigma \in 2^n$ we have that $\sigma \notin T_x$. Put $f(x) = n$.

(3) $\to$ (1). Input x. Compute a use bound u_x so that for every possible computation $\{e\}^?_{f(x)}(x)$ the computation only asks about $i < u_x$. (Since it takes at least one step to ask the oracle anything there are at most $2^{f(x)}$ such simulations.)

Now define

$$t_x = \{R \subseteq [0, u_x] \ : \ \{e\}^R_{f(x)}(x) \downarrow = 1\}.$$

Define

$$\theta_x = \bigvee_{R \in t_x} (\bigwedge_{i \in R} P_i \ \wedge \ \bigwedge_{i \in [0,u_x] \setminus R} \ \neg P_i)$$

Then for any $x \in \omega$ we have that

$$x \in A \text{ iff } \{e\}^B_{f(x)}(x) \downarrow = 1 \text{ iff } B \cap [0, u_x] = R \in t_x \text{ iff } B \models \theta_x.$$

QED

Proposition 34.13 *(Post)*

1. *If A is simple, then $A <_m K$.*
2. *If A is hypersimple, then $A <_{tt} K$.*
3. *There exists a simple A with $A \equiv_{tt} K$.*

Proof

(1) If $K \leq_m A$ then A is creative and hence not simple. (See Theorem 14.3.)

(2) Since every r.e. set is many-one reducible to K it is enough to see that $K \leq_{tt} A$ implies A is not hypersimple.

Claim. Let $\Gamma = \{P_n : n \in A\}$. Then there exists a recursive list $(\rho_n : n < \omega)$ of propositional sentences such that for every n

1. $A \models \rho_n$ and
2. $\Gamma \cup \{\rho_m : m < n\} \not\models \rho_n$.

Proof
Since $\overline{K} \leq_{tt} A$ there exists a recursive function $\theta : \omega \to SENT$ such that $n \in \overline{K}$ iff $A \models \theta(n)$.

Now we effectively construct ρ_n as follows. Let

$$\Sigma_n = \{\rho : \Gamma \cup \{\rho_m : m < n\} \vdash \rho\}.$$

Note that Σ_n is recursively enumerable as a subset of SENT. Also $A \models \theta$ for every $\theta \in \Sigma_n$. It follows that $\theta^{-1}(\Sigma_n) \subseteq \overline{K}$ is r.e. By the S-n-m Theorem there exists a recursive function f such that

$$W_{f(n)} = \theta^{-1}(\Sigma_n)$$

and by the proof that K is creative we have that

$$f(n) \in \overline{K \cup \theta^{-1}(\Sigma_n)}.$$

Take $\rho_n = \theta(f(n))$.
QED

Let S_k be that set of all n such that the propositional letter P_n occurs in the sentence ρ_k, i.e., S_k is the support of ρ_k.

Claim. For any n let

$$m = \max\left(\bigcup\{S_k \;:\; k \leq 2^{2^{n+1}} + 1\}\right)$$

then $\overline{A} \cap [n, m) \neq \emptyset$.
Proof
Suppose not and assume that $[n, m) \subseteq A$. Let ρ_k^* be obtained from ρ_k by replacing all propositional letters P_i for $n < i < m$ by the letter P_n. Note that $\Gamma \vdash P_i$ for all these i and hence $\Gamma \vdash \rho_k^* \equiv \rho_k$ for every $k \leq 2^{2^{n+1}} + 1$. But there are at most $2^{2^{n+1}}$ logically inequivalent propositional sentences with atomic letters P_i for $i \leq n$ and so for some $k < l$ we have that $\rho_k^* \equiv \rho_l^*$. But this is a contradiction since then

$$\Gamma \vdash \rho_i \equiv \rho_j.$$

QED

Now it is an easy matter to construct a recursive sequence $n_k < n_{k+1}$ so that $\overline{A} \cap [n_k, n_{k+1}) \neq \emptyset$ for each k. Hence A is not hypersimple.

(3) Let B be any simple set which is not hypersimple. By Proposition 34.8 there exists a recursive increasing sequence $(n_k : k < \omega)$ such that for all k we have that $\overline{B} \cap [n_k, n_{k+1}) \neq \emptyset$. Now let

$$A = B \cup \bigcup_{k \in K} [n_k, n_{k+1})$$

A is simple because it is a superset of the simple set B. $\overline{A}$ is infinite because for each $k \in \overline{K}$ we have $\overline{A} \cap [n_k, n_{k+1}) \neq \emptyset$. We have that $K \leq_{tt} A$ because

$$k \in K \text{ iff } A \models \bigwedge_{n_k \leq i < n_{k+1}} P_i$$

QED

Exercise 34.14. Prove that $\leq_{tt}$ is transitive, i.e., $A \leq_{tt} B$ and $B \leq_{tt} C$ implies $A \leq_{tt} C$.

35 Hyperhypersimple sets

Definition 35.1 *V is a weak array iff V is r.e. and $V_x \cap V_y = \emptyset$ whenever $x \neq y$. As usual, $V_x = \{y : \langle x, y \rangle \in V\}$.*

Definition 35.2 *$A \subseteq \omega$ is hyperhypersimple iff A is re, $\overline{A}$ is infinite, and for every weak array V there exists x with $V_x \subseteq A$.*

Proposition 35.3 *For any $A \subseteq \omega$ for which A is r.e. and $\overline{A}$ is infinite the following are equivalent:*

1. *A is hyperhypersimple*
2. *for every infinite r.e. set B such that $W_x \cap W_y = \emptyset$ for all distinct $x, y \in B$ there exists $x \in B$ with $W_x \subseteq A$*
3. *for every weak array V there exists an infinite recursive set R such that $V_x \subseteq A$ for all $x \in R$*
4. *for every weak array V such that V_x is finite for all x there exists x such that $V_x \subseteq A$*

Proof

(1) iff (2) is true because the two types of arrays are the same.

(1) $\to$ (3), The sequence $(R_n = \{\langle n, m\rangle : m \in \omega\} : n < \omega)$ is a uniformly recursive partition of ω into infinite pieces. Take

$$U_n = \cup_{e\in R_n} V_e$$

Then U is weak array and so there exists n with $U_n \subseteq A$.

(4) $\to$ (1). Given a weak array V such that $V_e \cap \overline{A} \neq \emptyset$ for all e we find another weak array V^* such that V_e^* finite and $V_e^* \cap \overline{A} \neq \emptyset$ for all e. For each s define $V_{e,s}^* = V_{e,s_0+1}$ where s_0 is the largest $t \leq s$ such that $V_{e,t} \subseteq A_s$.
QED

Exercise 35.4. Prove

(a) If A is simple and B is simple, then $A \cap B$ is simple.

(b) If A is hypersimple and B is hypersimple, then $A \cap B$ is hypersimple.

(b) If A is hyperhypersimple and B is hyperhypersimple, then $A \cap B$ is hyperhypersimple.

Example 35.5 *There exists a hypersimple set A which is not hyperhypersimple.*

Proof
Let $B \subseteq \omega$ be any hypersimple set. Define $A \subseteq \omega$ by

$$A = \{\langle n, m\rangle : n \in B \text{ or } n \leq m\}.$$

See Figure 6. A is not hyperhypersimple since each of the horizontal lines:

$$V_k =^{def} \{\langle m, k\rangle \;:\; m \in \omega\}$$

meets $\overline{A}$. To see that A is hypersimple suppose we are given a strong array $(D_n : n \in R)$. Let $\pi(\langle m, n\rangle) = m$ be projection to the first coordinate. We can find an infinite recursive subset $S \subseteq R$ such that $(\pi(D_x \cap Q) : x \in S)$ are pairwise disjoint where $Q = \{\langle n, m\rangle : m < n < \omega\}$. Since B is hypersimple, there exists $x \in S$ with $\pi(D_x \cap Q) \subseteq B$ and hence $D_x \subseteq A$.
QED

Example 35.6 *Dekker deficiency sets are never hyperhypersimple.*

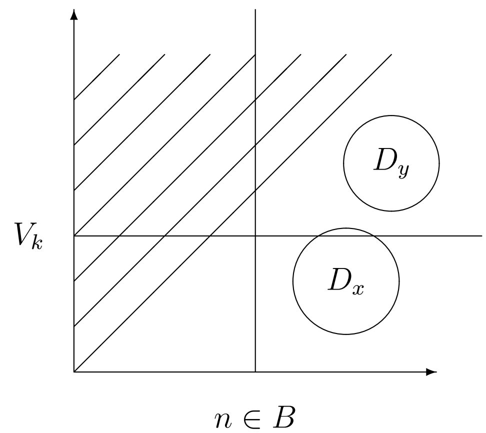


Figure 6: $A = \{\langle n, m\rangle : n \in B \text{ or } n \leq m\}$.

Proof
Let $A = \{a_s : s \in \omega\}$ be a one-one recursive enumeration of a non recursive set A. And $D = \{s : \exists t > s \;\; a_t < a_s\}$. We construct a weak array V to meet the requirements:

$$R_x \qquad V_x \cap \overline{D} \neq \emptyset$$

Stage s+1

Step (a). For any $x \leq s$ if R_x has a follower t such that $a_s < a_t$ then unappoint t so that now R_x has no follower.

Step(b). For the least x for which R_x has no follower, appoint s the follower of R_x and put $V_{x,s+1} = V_{x,s} \cup \{s\}$.

This ends the stage and the construction. Note that V is a weak array.

Claim. Each R_x obtains a permanent follower s and for this s we have $s \in V_x \cap \overline{D}$.

Proof
This is by induction on x. So after some sufficiently large stage s_0 no $y < x$ is appointed a new follower. Suppose for contradiction that R_x is appointed a new follower at stages $s_1, s_2, \ldots$ where $s_0 < s_1 < s_2 < \cdots$. Note that since higher priority requirements don't get new followers after s_0 each time R_x

losses its follower it acquires the stage itself as its new follower. But this means that

$$a_{s_1} > a_{s_2} > a_{s_3} > \cdots$$

which is a contradiction.
QED

Exercise 35.7. Prove that if A is hypersimple and $(D_x : x \in C)$ is a strong array then there exists an infinite computable $E \subseteq C$ such that $D_x \subseteq A$ for all $x \in E$.

36 Maximal sets

Definition 36.1 *$A \subseteq^* B$ iff $B \setminus A$ is finite.*
$A =^* B$ iff $A \subseteq^* B$ and $B \subseteq^* A$
$\forall^\infty$ means 'for all but finitely many'
$\exists^\infty$ means 'exists infinitely many'

Definition 36.2 *$M \subseteq \omega$ is maximal iff M is r.e. , $\overline{M}$ is infinite, and for every A r.e. if $M \subseteq A$ then $M =^* A$ or $A =^* \omega$.*

Proposition 36.3 *Maximal sets are hyperhypersimple.*

Suppose V is a weak array such that $V_e \cap \overline{A} \neq \emptyset$ for all e. Define

$$B = A \cup \bigcup_{e<\omega} V_{2e}$$

then $A \neq^* B$ and $B \neq^* \omega$, so A is not maximal.
QED

Theorem 36.4 *(Friedberg) Maximal sets exist.*

Proof
We will construct the maximal set M as follows. We use the notation p_n for the n^{th} element of the complement of M, i.e.,

$$\overline{M} = \{p_0 < p_1 < p_2 < \cdots\}$$

Are requirements are

$R_e \qquad (\forall^\infty n \;\; p_n \in W_e) \quad \text{or} \quad (\forall^\infty n \;\; p_n \notin W_e)$

This guarantees that $M \cup W_e =^* \omega$ or $M \cup W_e =^* M$,

At stage s given M_s we let

$$\overline{M_s} = \{p_{0,s} < p_{1,s} < p_{2,s} < \cdots\}$$

The idea of this proof is called moving markers. We think of a marker labeled n with position $p_{n,s}$. As we slide the marker upward we put the uncovered numbers into M_s. In order to get $\overline{M}$ infinite we want each marker to eventually stop moving.

Definition 36.5 *$\sigma \in 2^n$ is the n-state of x at stage s iff*

$$\textit{for all } e < n \qquad \sigma(e) = \begin{cases} 1 & \textit{if } x \in W_{e,s} \\ 0 & \textit{if } x \notin W_{e,s} \end{cases}$$

Two easy facts about the n-state are the following:

(1) Suppose $s_1 \leq s_2$,
$\sigma_1 \in 2^n$ is the n-state of x at stage s_1, and
$\sigma_2 \in 2^n$ is the n-state of x at stage s_2,
then $\sigma_1 \leq_{lex} \sigma_2$.

(2) For fixed n and x there is $\sigma \in 2^n$ such that σ is the n-state of x for all but finitely many stages s. We call this the final n-state of x.

Our strategy can be summarized simply as 'maximize the lexicographic order of the n-state of p_n'.

Stage $s+1$.
Find the least n (if any) such that there exists m with $n < m < s$ such that
if $\sigma \in 2^n$ is the n state of $p_{n,s}$ and
$\tau \in 2^n$ is the n state of $p_{m,s}$, then $\sigma <_{lex} \tau$.
For the least such n find the least m and shift the marker n to m:
Put $p_{n+i,s+1} = p_{m+i,s}$ for all $i < \omega$. Equivalently put

$$M_{s+1} = M_s \cup \{p_{j,s} : n \leq j < m\}$$

Otherwise as usual if there are no such n, m just go to the next stage with everything unchanged.

This ends the stage and the construction.

Claim. The markers eventually stop moving, i.e.,

$$\lim_{s\to\infty} p_{n,s} = p_n < \infty$$

Proof
This is proved by induction on n. Note that the only way the marker n moves is either that it is bumped up by some marker $m < n$ or it moves to a higher n-state. So consider some stage s_0 so that no marker $m < n$ moves after stage s_0. But it is impossible for p_n to change infinitely many times after this since its n-state would have to increase lexicographically infinitely many times. (Note that in between moves its n-state might also change without the marker moving but it can only increase if it doesn't move.)
QED

Claim. For each n there exists $\tau \in 2^n$ such that

$$\forall^\infty m \;\; \tau = \text{ the final } n\text{-state of } p_m.$$

Proof
Suppose not. Then there exists distinct $\tau_1, \tau_2 \in 2^n$ such that
$\exists^\infty m \;\; \tau_1 =$ the final n-state of p_m and
$\exists^\infty m \;\; \tau_2 =$ the final n-state of p_m.
Suppose $\tau_1 <_{lex} \tau_2$. Then we can choose m_1, m_2 with $n < m_1 < m_2$ and the final n-state of p_{m_i} is τ_i. This is a contradiction, since for some large enough stage $s_0 > m_2$ the markers p_j for $j \leq m_2$ have stopped moving and their final n-states are their states at stage s_0. But by the construction some marker $\leq p_{m_1}$ must move.
QED

This final claim proves the Theorem, since if $n = e+1$ we have that
$\tau(e) = 1$ implies $\forall^\infty m \;\; p_m \in W_e$
and
$\tau(e) = 0$ implies $\forall^\infty m \;\; p_m \notin W_e$
QED

Example 36.6 *There exists a hyperhypersimple set which is not maximal.*

Proof
First we note that it easy to get M_1 and M_2 maximal so that $M_1 \neq^* M_2$. Take any maximal set M and let $R \subseteq M$ to be an infinite recursive subset. Let $\pi : \omega \to \omega$ be a recursive bijection which takes R to $\overline{R}$. Let $M_1 = M$ and let $M_2 = \pi(M_1)$.

Now let $A = M_1 \cap M_2$. Then A is hyperhypersimple (see exercise) but not maximal since $A \subseteq M_1 \subseteq \omega$ and $A \neq^* M_1$ and $M_1 \neq^* \omega$.
QED

Remark. Yates noted that we can add to the maximal set construction an extra 'kick' to the p_e marker to ensure that $\{e\}(e)\downarrow$ iff $\{e\}_{p_e}(e)$. Then the maximal set constructed will be Turing equivalent to K.

Exercise 36.7. Suppose $A = \{a_n : n < \omega\}$ is a 1-1 recursive enumeration of a hyperhypersimple set A. Let $B = \{a_{a_n} : n < \omega\}$. Prove that B is hyperhypersimple but not maximal.

Exercise 36.8. An r.e. set $A \subseteq \omega$ is simple in R where R is an infinite recursive set iff $\overline{A} \cap R$ is infinite but contains no infinite r.e. subset. Is every r.e. set which is not recursive, simple in some infinite recursive set? Hint: Split a maximal set.

37 The lattice of r.e. sets

Definition 37.1 *The lattice of r.e. sets is $\mathcal{E} = (r.e.sets, \subseteq)$. A subset $X \subseteq \mathcal{E}$ is definable iff there is a first order formula $\theta(v)$ in the language of $\subseteq$ such that*

$$X = \{A \in \mathcal{E} \ : \ \mathcal{E} \models \theta(A)\}.$$

Similarly for $X \subseteq \mathcal{E}^2$ or $X \subseteq \mathcal{E}^3$.

Example 37.2 *The following are definable in $\mathcal{E}$.*

1. $\{(A, B, C) \in \mathcal{E}^3 \ : \ A \cup B = C\}$
2. $\{(A, B, C) \in \mathcal{E}^3 \ : \ A \cap B = C\}$
3. $\{\emptyset\}$
4. $\{\omega\}$

5. *recursive sets*

   A is recursive iff $\mathcal{E} \models \exists B \ \ B \cap A = \emptyset$ *and* $B \cup A = \omega$

6. *r.e. but not recursive sets*

7. *infinite r.e. sets*

   A is infinite r.e. iff $\mathcal{E} \models \exists B \ \ B \subseteq A$ *and B is not recursive*

8. *finite sets*

9. *cofinite sets*

10. $\subseteq^*$, $=^*$

11. *simple sets*

12. *maximal sets*

Definition 37.3 *π is an automorphism of $\mathcal{E}$ iff $\pi : \mathcal{E} \to \mathcal{E}$ is a bijection such that for every $A, B \in \mathcal{E}$*

$$A \subseteq B \text{ iff } \pi(A) \subseteq \pi(B).$$

Note that for any first-order formula $\theta(v_1, \ldots, v_n)$ in the language of $\mathcal{E}$, i.e., $\subseteq$, that for any $\pi \in aut(\mathcal{E})$ and $A_1, \ldots, A_n \in \mathcal{E}$ we have that

$$\mathcal{E} \models \theta(A_1, \ldots, A_n) \text{ iff } \mathcal{E} \models \theta(\pi(A_1), \ldots, \pi(A_n))$$

Hence definable sets are closed under automorphisms.

Example 37.4 *If $A \in \mathcal{E}$, then $\{A\}$ is definable in $\mathcal{E}$ iff $A = \emptyset$ or $A = \omega$.*

Proof
If A is neither $\emptyset$ or ω, then we can choose $n, m < \omega$ such that $n \in A$ and $m \notin A$. Let $\pi : \omega \to \omega$ be the identity except $\pi(n) = m$ and $\pi(m) = n$. Define $\pi : P(\omega) \to P(\omega)$ by $\pi(A) = \{\pi(n) : n \in A\}$. Then since π is recursive it is clear that $\pi \in aut(\mathcal{E})$. But since

$$\pi(A) = (A \setminus \{n\}) \cup \{m\}$$

we see that $\{A\}$ is not closed under automorphisms and hence cannot be definable.
QED

Proposition 37.5 *1. For every $\pi \in aut(\mathcal{E})$ there exists a bijection $\hat{\pi}$ of ω such that $\pi(A) = \{\hat{\pi}(n) : n \in A\}$.*

2. Not every bijection $\pi : \omega \to \omega$ induces an automorphism of $\mathcal{E}$.

3. There are continuum many bijections $\pi : \omega \to \omega$ which induce an automorphism of $\mathcal{E}$.

Proof

(1) It is easy to see that the set of singletons

$$\{\{n\} : n \in \omega\} \subseteq \mathcal{E}$$

is definable in $\mathcal{E}$. Hence any automorphism $\pi : \mathcal{E} \to \mathcal{E}$ must permute the singletons. Define $\hat{\pi}(n)$ so that $\pi(\{n\}) = \{\hat{\pi}(n)\}$. But now for every $n \in \omega$

$$n \in A \text{ iff } \{n\} \subseteq A \text{ iff } \pi(\{n\}) \subseteq \pi(A) \text{ iff } \hat{\pi}(n) \in \pi(A)$$

Hence $\pi(A) = \{\hat{\pi}(n) : n \in A\}$.

(2) Take any bijection which maps the even integers to some non recursive infinite coinfinite set.

(3) Let M be a maximal set. Let $\pi : \omega \to \omega$ be any bijection such that $\pi \restriction M = id$. There are continuum many such bijections, one for each permutation of $\overline{M}$. But for any $A \in \mathcal{E}$ we have that $A \cap \overline{M}$ is finite or $A \cap \overline{M} =^* \overline{M}$. But this gives us that $\pi(A) =^* A$. Similarly $\pi^{-1}(A) =^* A$.
QED

The following theorem shows that the family of hyperhypersimple sets is definable in $\mathcal{E}$.

Theorem 37.6 *(Lachlan) A is hyperhypersimple iff A is r.e. , $\overline{A}$ is infinite, and*

$$\mathcal{E} \models \forall B \supseteq A \;\; \exists C \supseteq A \;\; B \cap C = A \text{ and } B \cup C = \omega$$

Proof

Suppose A is not hyperhypersimple and V is a weak array such that $V_e \cap \overline{A} \neq \emptyset$ for all e. Define

$$B = A \cup \bigcup_{e \in \omega} (V_e \cap W_e)$$

Suppose for contradiction that C satisfies $B \cap C = A$ and $B \cup C = \omega$. Then for some e we have that $C = W_e$. Let $x \in V_e \cap \overline{A}$. If $x \in W_e$ then $x \in C \cap B$

but this contradicts $B \cap C = A$. If $x \notin W_e$ then $x \notin C$ and $x \notin B$ but this contradicts $B \cup C = \omega$.

Conversely suppose there exists B as above for which there is no C. We must show there is a weak array V such that $V_e \cap \overline{A} \neq \emptyset$ for all e. So let $B = \{b_s : s \in \omega\}$ be a 1-1 recursive enumeration of B and put $B_s = \{b_t : t < s\}$. Similarly, let A_s be a recursive enumeration of A.

We will construct $V_{e,s}$ pairwise disjoint subsets of B and meet the requirements:

$$R_e \qquad V_e \cap \overline{A} \neq \emptyset$$

We will carry along $g(e,s)$ a gate which we use to let elements into each V_e. At stage $s = 0$ as usual we put $V_{e,s} = \emptyset$ and $g(e,s) = 0$.

Construction

Stage $s + 1$**.**
First define $g(e,s+1)$ for $e < s$. If $V_{e,s} \subseteq A_s$, then $g(e,s+1) = g(e,s) + 1$. In other words, if the requirement R_e is not looking good, then increment the gate, otherwise let it alone.

Look for the least $e < s$ (if any) such that $b_s \leq g(e,s+1)$ and put b_s into V_e, i.e.,

$$V_{e,s+1} = V_{e,s} \cup \{b_s\}.$$

If there is no such e, do nothing. This ends the construction.

Verification

Claim. $\lim_{s\to\infty} g(e,s) = g(e) < \infty$ and R_e is met.
Proof
This is proved by induction on e. Choose s_0 so that for all $\hat{e} < e$ and $s > s_0$ we have that $g(\hat{e},s) = g(\hat{e})$ and

$$b_s > \max\{g(\hat{e}) : \hat{e} < e\}$$

Suppose for contradiction that

$$\lim_{s\to\infty} g(e,s) = \infty$$

Define

$$C = A \cup \bigcup_{s \geq s_0} ([0, g(e, s+1)] \cap \overline{B_s}$$

Suppose $x \in \overline{A}$. Then we claim that

$$x \in C \text{ iff } x \in \overline{B}$$

This is a contradiction since then $C \cap B = A$ and $C \cup B = \omega$.

Suppose $x \in \overline{B}$. This implies that $x \in \overline{B_s}$ for all s. But if $g(e, s) \to \infty$ we have that $x \in C$.

Suppose $x \in C$. Then for some $s \geq s_0$ we have that $x \in [0, g(e, s)] \cap \overline{B_s}$ (since we are assuming $x \notin A$.) If $x \notin \overline{B}$ then $x \in B \setminus B_s$. Hence $x = b_t$ for some $t \geq s$. But notice that $b_t = x \leq g(e, s) \leq g(e, t+1)$. By our choice of s_0 we have that $b_t > g(\hat{e})$ for all $\hat{e} < e$ and so b_t will be put into V_e. But $x = b_t$ was assumed to be an element of $\overline{A}$. This means that $g(e, t)$ will never increase again which contradicts it going to ∞.

The reason R_e is met is because if $g(e, s)$ stops growing then eventually we stop putting b_s's into V_e. Hence V_e is finite and so it is impossible that $V_e \subseteq A$.

This proves the Claim and the Theorem.

QED

The following shows that the family of hypersimple sets is not definable in $\mathcal{E}$.

Theorem 37.7 *(Martin) There exists a hypersimple set A and $\pi \in aut(\mathcal{E})$ such that $\pi(A)$ is not hypersimple.*

Proof

We will construct the r.e. set A as usual by constructing a recursive increasing sequence A_s. We will construct a recursive sequence π_s of bijections of ω with the property that $\pi_s(n) = n$ for every $n \geq s$. So each π_s is really a finite permutation. π will be the limit of π_s.

Let $W^*_{e,s}$ be defined as follows:

$W^*_{e,s} = W_{e,s_0}$ where $s_0 \leq s$ is the largest $t \leq s$ with the property that for distinct $x, y \in W_{e,t}$ we have that $D_x \cap D_y = \emptyset$.

The list W^*_e automatically contains all strong arrays. Our requirements for this construction include:

$R_e \quad W^*_e$ infinite $\to \exists x \in W^*_e \quad D_x \subseteq A$

The strategy for making sure that $\overline{A}$ is a variant on the Post $2e$ strategy. At stage $s = 0$ in our construction we have $A_s = \emptyset$ and π_s the identity.

Stage $s + 1$**.**
Given π_s and A_s, we say that $e < s$ requires attention iff

1. $\neg\exists n \in W^*_{e,s} \quad D_n \subseteq A_s$
2. $\exists x, y$ such that
   (a) $x, y \notin A_s$
   (b) $\exists n \in W^*_{e,s} \;\; x \in D_n$
   (c) $e < x < y < s, \quad e < \pi_s(x), \quad e < \pi_s(y)$
   (d) i. e-state of x at stage s = e-state of y at stage s
       ii. e-state of $\pi_s(x)$ at stage s = e-state of $\pi_s(y)$ at stage s
   (e) $2x < \pi_s(y)$.

The action at this stage is the following. For the least $e < s$ (if any) which requires attention we choose the least x for which there is a y and then we choose the least y. For this choice $(e, x, y) = (e_s, x_s, y_s)$ we
(a) put x into A, $A_{s+1} = A_s \cup \{x_s\}$
(b) put $\pi_{s+1} = \pi_s \circ swap(x, y)$ where $swap(x, y)$ refers to the transposition which interchanges x and y.
As usual if there is no e which requires attention we do nothing and go onto the next stage.

This ends the construction. Let Q denote the stages s where action takes place at stage $s + 1$. Then

$$A = \{x_s : s \in Q\}$$

We define

$$\pi(u) = \lim_{s\to\infty} \pi_s(u)$$

although at this point we have not proved that this limit always exists. Note the pointwise limit of 1-1 functions must be 1-1 where it is defined.

Note that for $s \in Q$ we have that $\pi_{s+1}(x_s) = \pi_s(y_s)$. Since x_s enters A we have (by $2a$) that x_s will never be a x_t or y_t latter. It follows that $\pi(x_s) = \pi_{s+1}(x_s)$. Hence

$$B =^{def} \{\pi_{s+1}(x_s) : s \in Q\} = \{\pi(x_s) : s \in Q\}$$

is well defined and r.e.

Claim (1) for any n we have that $|B \cap [0, 2n]| \leq n$.
Proof
Note that (by $2e$) we have that $\pi(x_s) = \pi_s(y_s) > 2x_s$. Since each x_s is distinct the Claim follows.
QED

As we have seen before this implies that B is not hypersimple. (Proposition 34.4).

Claim (2) $\lim_{s\to\infty} \pi_s(u) = \pi(u) < \infty$ for every u.
Proof
Fix s_0 so that $A \cap [0, u] = A_{s_0} \cap [0, u]$. Now the only way that $\pi_{s+1}(u) \neq \pi_s(u)$ for some $s > s_0$ is if $u = x_s$ or $u = y_s$. But in either case since $x_s < y_s$ and x_s enters A we have A changes in the interval $[0, u]$ which is a contradiction.
QED

We don't know yet that π is onto.

Claim (3) For each e

(a) R_e is met.

(b) $\exists s_0\ \forall s > s_0 \quad e_s > e$

Proof
This is proved by induction on e.

(a) We may suppose by induction that there exists s_0 such that $e_s \geq e$ for all $s > s_0$. Suppose R_e is not met. Then W_e^* is infinite and for all $n \in W_e^*$ we have that $D_n \cap \overline{A} \neq \emptyset$. For each $n \in W_e^*$ define

$$u_n = \min(D_n \cap \overline{A})$$

Since the D_n are pairwise disjoint all of the u_n are distinct. Note there exist $\sigma, \tau \in 2^e$ such that

$\exists^\infty n \in W_e^* \quad \sigma =$ final e-state of u_n and $\tau =$ final e-state of $\pi(u_n)$.

Choose u_n and u_m such that

1. $n, m \in W_e^*$
2. $e < u_n < u_m$
3. $2u_n < \pi(u_m)$

4. σ is the final e-state of u_n and u_m, and

5. τ is the final e-state of $\pi(u_n)$ and $\pi(u_m)$.

Increase s_0 (if necessary) so that not only is $e_s \geq e$ for all $s \geq s_0$ but also so that

1. $n, m \in W^*_{e,s_0}$

2. $u_n < u_m < s_0$ and $\pi(u_n) < s_0$ and $\pi(u_m) < s_0$

3. $\pi_s(u_n) = \pi(u_n)$ and $\pi_s(u_m) = \pi(u_m)$ all $s > s_0$

4. σ is the e-state of u_n and u_m at stage s_0,

5. τ is the e-state of $\pi(u_n)$ and $\pi(u_m)$ at stage s_0 and

6. $A_{s_0} \cap [0, u_m] = A \cap [0, u_m]$

Recall that we chose $u_n, u_m \in \overline{A}$. It is easy to check that e requires attention at stage s_0 and u_n and u_m witness this fact. But this means that u_n or some smaller u enters A. But this contradicts the condition that A does not change below u_m.

(b) Suppose that $e_s \geq e$ for all $s > s_0$ and R_e is met. If W^*_e is infinite, then for some $n \in W^*_e$ we have that $D_n \subseteq A$. But this will be seen at some stage and so e will not require attention after that. If W^*_e is finite, then suppose that

$$\cup\{D_n \ : \ n \in W^*_e\} \subseteq [0, N].$$

After we reach a stage $s > s_0$ where $A_s \cap [0, N] = A \cap [0, N]$, then e will never again require attention because then A would change beneath N.
QED

Claim (4) π is onto.
Proof
Given z choose s_0 so that $e_s > z$ for all $s \geq s_0$. If $\pi_{s_0}(u) = z$, then u will never be either x_s or y_s for any $s \geq s_0$. This is because we required (2c) that $\pi_s(x_s), \pi_s(y_s) > e_s$. Hence $\pi(u) = z$.
QED

Claim (5)
(a) $\forall C \in \mathcal{E} \quad \pi(C) \in \mathcal{E}$

(b) $\forall C\in\mathcal{E}\quad \pi^{-1}(C)\in\mathcal{E}$
Proof
(a) Fix s_0 so that for all $s>s_0$ we have that $e_s>e$. Then we show that

$$\pi(W_e)=\bigcup_{s>s_0}\pi_s(W_{e,s})$$

To see this first suppose $y\in\pi(W_e)$. Then there exists $x\in W_e$ with $\pi(x)=y$ but for all sufficiently large s we have that $x\in W_{e,s}$ and $\pi_s(x)=\pi(x)$ and thus $y\in\pi_s(W_{e,s})$.

To see the other inclusion, suppose that $y\in\pi_s(W_{e,s})$ for some $s>s_0$. We claim that for every $t>s$ that $y\in\pi_t(W_{e,t})$. This is proved by induction on t. Suppose that $\pi_t(u)=y$ for some $u\in W_{e,t}$. Then $\pi_{t+1}(u)=\pi_t(u)$ unless $u=x_t$ or $u=y_t$ and then $\pi_{t+1}(x_t)=\pi_t(y_t)$ and $\pi_{t+1}(y_t)=\pi_t(x_t)$. But since x_t and y_t have the same e_t-state and $e_t>e$, if one is in $W_{e,t}$ so is the other. In either case we have that there exists $v\in W_{e,t+1}$ with $\pi_{t+1}(v)=y$. Now to see that $y\in\pi(W_e)$ suppose that $\pi(u)=y$ and choose sufficiently large $t>s_0$ such that $\pi_t(u)=\pi(u)=y$. Since π_t is a bijection and $y\in\pi_t(W_{e,t})$, it must be that $u\in W_{e,t}$ and hence $u\in W_e$.

(b) This is similar, except we use that $\pi_t(x_t)$ and $\pi_t(y_t)$ have the same e_t-state.
QED

Exercise 37.8. Prove that there exists a bijection $\pi:\omega\to\omega$ such that $\pi(A)\in\mathcal{E}$ for all $A\in\mathcal{E}$ but $\pi\notin aut(\mathcal{E})$. (Hint: use a maximal set.)

Exercise 37.9. For each $n\geq 2$ prove there is a sequence of maximal sets $A_1,\ldots,A_n$ such that $A_i\neq^* A_j$ for distinct i and j. Prove that for any such sequence that $\mathcal{E}^*(A_1\cap A_2\cap\cdots\cap A_n)$ is isomorphic to $(\mathcal{P}(\{1,\ldots,n\}),\subseteq)$. The structure $\mathcal{E}^*(A)$ is the set of r.e. supersets of A modulo the finite sets and ordered by $\subseteq^*$.

38 Arithmetic hierarchy

Definition 38.1 *For A and B predicates over subsets of ω or finite products of ω we define:*
$\Pi^0_0=\Sigma^0_0=$ *the recursive predicates.*
A is Σ^0_{n+1} iff there exists B which is Π^0_n and $A(x)\equiv\exists y\ B(x,y)$.

A is Π^0_{n+1} iff there exists B which is Σ^0_n and $A(x) \equiv \forall y \;\; B(x,y)$.
A is Δ^0_n iff A is Σ^0_n and A is Π^0_n.

Note that by DeMorgan's Laws

$$\Pi^0_n = \{\neg A : A \in \Sigma^0_n\} \text{ and } \Sigma^0_n = \{\neg A : A \in \Pi^0_n\}$$

and hence

$$\Delta^0_n = \{\neg A : A \in \Delta^0_n\}.$$

Proposition 38.2 *Suppose Γ is Σ^0_n, Π^0_n, or Δ^0_n. Then Γ is closed under $\leq_m$, i.e., $A \leq_m B \in \Gamma$ implies $A \in \Gamma$. This implies that if the predicate $B(x,y)$ is in Γ and f is a recursive function, then $A(x,y) \equiv B(x, f(x))$ is in Γ.*

Proposition 38.3 *If $B(x,y)$ in Σ^0_n, then $A(x) \equiv \exists y \;\; B(x,y)$ is in Σ^0_n. If $B(x,y)$ in Π^0_n, then $A(x) \equiv \forall y \;\; B(x,y)$ is in Π^0_n.*

Proposition 38.4 *Suppose Γ is Σ^0_n, Π^0_n, or Δ^0_n. If $A, B \in \Gamma$, then $A \wedge B$ and $A \vee B$ are both in Γ. Also, Γ predicates are closed under bounded quantification, e.g., $\exists u < x \; A(u, x, \ldots)$ and $\forall u < x \; A(u, x, \ldots)$.*

Proposition 38.5 $\Sigma^0_n \cup \Pi^0_n \;\subseteq\; \Delta_{n+1} \;=\; \Sigma^0_{n+1} \cap \Pi^0_{n+1}$

Definition 38.6 *We say that A is universal for Γ iff*

$$\Gamma = \{A_x \;:\; x \in \omega\}.$$

We say that A is m-complete for Γ iff

$$\Gamma = \{B \;:\; B \leq_m A\}$$

Note that universal for Γ implies m-complete for Γ. Also, the complement of a set universal for Γ is universal for $\widetilde{\Gamma}$ and the same for m-completeness.

Proposition 38.7 *For each $n > 0$ there is a universal Σ^0_n set.*

Proposition 38.8 *For each $n > 0$ we have $Red(\Sigma^0_n)$, $Sep(\Pi^0_n)$, $\neg Sep(\Sigma^0_n)$, and $\neg Red(\Pi^0_n)$.*

Proof
See definitions 8.2. We first show $Red(\Sigma^0_n)$. Let

$$A(x) \equiv \exists y\ R(x,y) \quad \text{and} \quad B(x) \equiv \exists y\ S(x,y)$$

where R and S are Δ^0_n. Reduce A and B by

$$A_0(x) \equiv \exists y\ (R(x,y) \wedge \forall z < y\ \neg S(z,x))$$

and

$$B_0(x) \equiv \exists y\ (S(x,y) \wedge \forall z \leq y\ \neg R(z,x))$$

Since $Red(\Gamma) \to Sep(\widetilde{\Gamma})$ Proposition 8.4, it follows that $Sep(\Pi^0_n)$ holds.

To see $\neg Sep(\Sigma^0_n)$, first construct a doubly universal pair A and B. These are Σ^0_n sets such that for every pair C and D of Σ^0_n sets there exists a u with $C = A_u$ and $D = B_u$. To get A and B let U be a universal Σ^0_n set. Then define

$$A = \{(\langle x,y\rangle, z)\ :\ \langle x,z\rangle \in U\}$$

and

$$B = \{(\langle x,y\rangle, z)\ :\ \langle y,z\rangle \in U\}$$

then $u = \langle x,y\rangle$ codes the pair U_x and U_y. Now applying reduction to A and B we get $A^0 \subseteq A$ and $B^0 \subseteq B$. Note that this simultaneously reduces all cross sections A_u and B_u. Assuming for contradiction that separation holds, let C be Δ^0_n such that $A^0 \subseteq C$ and $B^0 \subseteq \overline{C}$. We get a contradiction since, then C would be a universal Δ^0_n set. This is because if P is Δ^0_n then there exists u with $A_u = P$ and $B_u = \overline{P}$. But the reduction followed by separation can't effect the u cross section, so $C_u = P$.
QED

Exercise 38.9. Prove there does not exist a universal Δ^0_n set.

39 Post: Δ^0_2 same as recursive in $0'$

Lemma 39.1 *$A \subseteq \omega$ is Π^0_2 iff there exists P recursive such that*

$$A(x) \text{ iff } \exists^\infty s\ \ P(s,x)$$

Proof
($\leftarrow$) $\exists^\infty s\ \ P(s,x)$ iff $\forall t\ \exists s > t\ P(s,x)$
($\rightarrow$) Suppose

$$A(x) \text{ iff } \forall n\ \exists m\ R(n,m,x)$$

where R is Δ^0_1. Define $P \subseteq \omega^{<\omega} \times \omega$ by

$$P(\sigma,x) \text{ iff } \forall i < |\sigma|\ \ [R(i,\sigma(i),x) \text{ and } \forall j < i\ \neg R(i,j,x)]$$

QED

Theorem 39.2 *(Post) Suppose $A \subseteq \omega$. Then A is Δ^0_2 iff $A \leq_T 0'$*

Proof
Suppose A is Δ^0_2. Then by Lemma 39.1 there exists recursive $P(u,x)$ and $Q(v,x)$ such that

$$A(x) \equiv \exists^\infty u\ \ P(u,x)$$

$$\neg A(x) \equiv \exists^\infty v\ \ Q(v,x)$$

Now define $g(x,s)$ as follows. Input x,s and let u_s be the maximum $u \leq s$ such that $P(u,x)$ (zero if no such u). Similarly define v_s to be the maximum $v \leq s$ such that $Q(v,x)$. Define

$$g(x,s) = \begin{cases} 1 & \text{if } u_s \geq v_s \\ 0 & \text{if } u_s < v_s \end{cases}$$

It is easy to check that

$$A(x) = \lim_{s\to\infty} g(x,s)$$

and so by the Limit Lemma 28.1 we have that $A \leq_T 0'$.

Conversely if $A \leq_T 0'$ then by the Limit Lemma we have g recursive such that

$$A(x) = \lim_{s\to\infty} g(x,s)$$

but then

$$A(x) \equiv \forall^\infty s\ \ g(x,s) = 1 \equiv \exists^\infty s\ \ g(x,s) = 1$$

so A is Δ^0_2.
QED

Lemma 39.3 *(1) $A \subseteq \omega$ is $\Sigma^0_1(B)$ iff $A \leq_m B'$.*
(2) A is $\Delta^0_2(B)$ iff $A \leq_T B'$.

Proof
A is $\Sigma^0_1(B)$ iff there exists a predicate $R \leq_T B$ such that

$$A(x) \text{ iff } \exists y \;\; R(x,y)$$

(1) is just a relativization of the standard result that $0'$ is m-complete for Σ^0_1.
(2) is just the relativization of Post's Theorem 39.2.
QED

Theorem 39.4 *(Post)*
(1) $A \leq_T 0^{(n)}$ *iff* A *is* Δ^0_{n+1}.
(2) $0^{(n)}$ *is an m-complete* Σ^0_n*-set.*

Proof
(1) for $n = 2$:
$A \leq_T 0''$ iff $A \leq_T (0')'$ iff A is $\Delta^0_2(0')$.
A is $\Delta^0_2(0')$ iff there exists $R_1, R_2 \leq_T 0'$ such that

$$A(x) \text{ iff } \exists n \, \forall m \; R_1(n,m)$$

$$\neg A(x) \text{ iff } \exists n \, \forall m \; R_2(n,m)$$

but since $R_1, R_2 \leq_T 0'$ iff R_1 and R_2 are Δ^0_2, we have that
A is $\Delta^0_2(0')$ iff A is Δ^0_3.
(2) for $n = 2$:
$0''$ is $\Sigma^0_1(0')$ and m-complete for $\Sigma^0_1(0')$. But $\Sigma^0_1(0')$ is Σ^0_2. This is because B is $\Sigma^0_1(0')$ iff there exists $R \leq_T 0'$ such that

$$B(x) \text{ iff } \exists y \;\; R(x,y)$$

But $R \leq_T 0'$ iff R is Δ^0_2. Hence B is Σ^0_2 iff B is $\Sigma^0_1(0')$.

The proofs for $n > 2$ are analogous.
QED

Exercise 39.5. Prove there does not exist A which is m-complete for Δ^0_2.

Exercise 39.6. (Enderton, Putnam) Prove that if $0^{(n)} \leq_T A$ for every n, then $0^{(\omega)} \leq_T A''$.
Hint: Show that

$$P(e_1, e_2) \equiv \exists B, C \quad B = \{e_1\}^A \wedge C = \{e_2\}^A \wedge C = B'$$

is $\Pi^0_2(A)$.

40 EMP, TOT, FIN, and REC

Proposition 40.1 *$EMP =^{def} \{e : W_e = \emptyset\}$ is Π^0_1-m-complete.*

Proof

$$e \in EMP \text{ iff } \forall x, s \;\; x \notin W_{e,s}$$

so EMP is Π^0_1. Let A be Π^0_1, then there is R recursive so that

$$A(x) \text{ iff } \forall y \;\; R(x,y).$$

Using S-n-m Theorem get f recursive so that for every x

$$W_{f(x)} = \{y : \neg R(x,y)\}$$

Then $A(x)$ iff $f(x) \in EMP$.
QED

Proposition 40.2 *$TOT =^{def} \{e : W_e = \omega\}$ is m-complete for Π^0_2.*
$FIN =^{def} \{e : W_e$ is finite$\}$ is m-complete for Σ^0_2.

Proof

$$e \in TOT \text{ iff } \forall x \, \exists s \; x \in W_{e,s}$$

$$e \in FIN \text{ iff } \exists x \, \forall y, s \; (y \in W_{e,s} \to y < x)$$

so TOT is Π^0_2 and FIN is Σ^0_2. Now suppose that A is Π^0_2 we show that

$$(A, \overline{A}) \leq_m (TOT, FIN)$$

which simultaneously shows that TOT is Π^0_2-m-complete and FIN is Σ^0_2-m-complete. Suppose

$$A(x) \text{ iff } \exists^\infty s \;\; P(s,x)$$

where P is Δ^0_1. Using S-n-m find a recursive function f so that

$$W_{f(x)} = \{t : \exists s > t \;\; P(s,x)\}$$

Hence $A(x) \to W_{f(x)} = \omega$ while $\neg A(x) \to W_{f(x)}$ is finite.
QED

Proposition 40.3 *$COF =^{def} \{e : \overline{W_e}$ is finite $\}$ is Σ^0_3-m-complete.*

Proof

$$e \in COF \text{ iff } \exists n\ \forall m > n\ \exists s\ \ m \in W_{e,s}$$

Now suppose that A is Σ^0_3. Then there exists P which is Δ^0_1 such that

$$A(x) \text{ iff } \exists n\ \ \exists^\infty m\ \ P(n, m, x)$$

Input x and describe the r.e. set B_x by using a moving marker construction similar to the construction of a maximal set but simpler. At any stage s we have that

$$\overline{B_{x,s}} = \{p_{0,s} < p_{1,s} < p_{2,s} < \cdots\}$$

We look for the least $n < s$ (if any) such that $P(n, s, x)$ and bump the n^{th} marker, i.e., enumerate $p_{n,s}$ into B_x, i.e., $B_{x,s+1} = B_{x,s} \cup \{p_{n,s}\}$. Note that if $A(x)$ is true then there exist n so that the n^{th} marker is bumped infinitely often and so B_x is cofinite. On the other hand if $\neg A(x)$, then each marker eventually stops moving and so B_x is coinfinite.

By the usual S-n-m argument we can find a recursive function f so that $B_x = W_{f(x)}$ for all x and so

$$A(x) \text{ iff } f(x) \in COF$$

QED

Proposition 40.4 *$REC =^{def} \{e\ :\ W_e$ is recursive $\}$ is Σ^0_3-m-complete.*

Proof

$$e \in REC \text{ iff } \exists e'\ \ (W_e \cup W_{e'} = \omega \text{ and } W_e \cap W_{e'} = \emptyset)$$

and $W_e \cup W_{e'} = \omega$ is Π^0_2 and $W_e \cap W_{e'} = \emptyset$ is Π^0_1. To see that it is m-complete, use a moving marker argument as above. Just add an additional reason to bump the e^{th} marker to make sure that if B_x is coinfinite, then for each e

$$\psi_e(e)\downarrow\ \rightarrow\ \ \psi_{e,p_e}(e)\downarrow$$

This guarantees that if B_x is coinfinite, then $K \leq_T B_x$.
QED

Exercise 40.5.

(a) Let A be an infinite r.e. set. Let

$$Q_A = \{e\ :\ W_e = A\}$$

Prove that Q_A is Π^0_2-m-complete.

(b) Let A be a finite nonempty set. Prove that

$$Q_A = \{e \;:\; W_e = A\}$$

is $D(\Sigma^0_1)$-m-complete, where

$$D(\Sigma^0_1) = \{A \cap \overline{B} \;:\; A, B \in \Sigma^0_1\}.$$

Lemma 40.6 *Suppose A is Σ^0_{k+1} then there exists B Π^0_k such that*

$$A(x) \textit{ iff } \exists y \;\; B(x,y) \textit{ iff } \exists ! y \;\; B(x,y)$$

Proof

Suppose

$$A(x) \text{ iff } \exists y \;\; P(x,y)$$

where P is Π^0_k. Then

$$A(x) \text{ iff } \exists y \;\; (P(x,y) \wedge \forall z < y \; \neg P(x,z)$$

Define

$$C(x,y) \text{ iff } \forall z < y \; \neg P(x,z)$$

In case $k+1 = 1$ then C is Δ^0_1. In case $k+1 > 1$ then since C is Σ^0_k we have by induction a Π^0_{k-1} predicate D so that

$$C(x,y) \text{ iff } \exists u \; D(x,y,u) \text{ iff } \exists ! u \; D(x,y,u)$$

Hence

$$A(x) \text{ iff } \exists y \; \exists u \;\; (P(x,y) \wedge D(x,y,u)) \text{ iff } \exists ! y \; \exists ! u \;\; (P(x,y) \wedge D(x,y,u))$$

so taking $B(x, \langle y,u \rangle) \equiv P(x,y) \wedge D(x,y,u)$ does the trick.

QED

Proposition 40.7 *(a) A is Π^0_3 iff there exists B which is Δ^0_1 such that*

$$A(u) \equiv \exists^\infty s \; \forall n \; B(s,n,u)$$

(b) A is Π^0_4 iff there exists B which is Δ^0_1 such that

$$A(x) \equiv \exists^\infty s \; \exists^\infty t \; B(s,t,x)$$

Proof

(a) Suppose

$$A(u) \equiv \forall x\ \exists y\ \forall z\ R(x,y,z,u)$$

where R is Δ^0_1. Define

$$Q(x,u) \equiv \exists y\ \forall z\ R(x,y,z,u)$$

Then by Lemma 40.6 there is a C which is Π^0_1 and

$$Q(x,u) \equiv \exists y\ \ C(x,y,u) \equiv \exists! y\ \ C(x,y,u)$$

Hence

$$A(u) \equiv \forall x \exists! y\ \ C(x,y,u)$$

$$A(u) \equiv \exists^\infty \sigma \in \omega^{<\omega}\ \ \forall i < |\sigma|\ \ C(i,\sigma(i),u)$$

Note that $\forall i < |\sigma|\ \ C(i,\sigma(i),u)$ is Π^0_1 and so there is B recursive so that

$$\forall n\ B(\sigma,n,u) \equiv \forall i < |\sigma|\ \ C(i,\sigma(i),u)$$

(b) Suppose

$$A(u) \equiv \forall x\ \exists y\ R(x,y,u)$$

where R is Π^0_2. By Lemma 40.6 applied to $\exists y\ \ R(x,y,u)$ we may assume that

$$A(u) \equiv \forall x\ \exists! y\ R(x,y,u)$$

Hence

$$A(u) \equiv \exists^\infty \sigma\ \forall i < |\sigma|\ \ R(i,\sigma(i),u)$$

but the predicate

$$Q(\sigma,u) \equiv \forall i < |\sigma|\ \ R(i,\sigma(i),u)$$

is Π^0_2 so there exists a recursive B so that

$$Q(\sigma,u) \equiv \exists^{<\infty} \tau\ \ B(\sigma,\tau,u)$$

Hence

$$A(u) \equiv \exists^\infty \sigma\ \exists^\infty \tau\ \ B(\sigma,\tau,u)$$

QED

Exercise 40.8. For the correct class Γ, show that INF, EQ, EQ* are m-complete Γ sets where

$$\text{INF} = \{e : W_e \text{ is infinite}\}$$

$$\text{EQ} = \{\langle e_1, e_2\rangle : W_{e_1} = W_{e_2}\}$$

and

$$\text{EQ}^* = \{\langle e_1, e_2\rangle : W_{e_1} =^* W_{e_2}\}.$$

Exercise 40.9.

Let $PTIME = \{e : \psi_e$ runs in polynomial time $\}$, i.e., there exists a polynomial $p(x)$ such that $\psi_e(x)$ halts in less than $p(x)$ steps for every x. Prove that $PTIME$ is Σ^0_2-m-complete.

Exercise 40.10. For each e let $Q_e = \{\frac{n}{m+1} : \langle n, m\rangle \in W_e\} \subseteq \mathbb{Q}$. Define

$\Omega = \{e : Q_e$ is order isomorphic to $\omega\}$.

Prove that Ω is Π^0_3-m-complete.

Exercise 40.11. Show that if $Q = \{e : W_e \in \mathcal{V}\}$ is not Σ^0_3 then $\mathcal{V}$ cannot be an r.e. class. See Definition 33.3.

Exercise 40.12. Prove that the family of coinfinite r.e. sets is not an r.e. class.

Exercise 40.13. Prove that SIMP=$\{e : W_e$ is simple $\}$ is m-complete Π^0_3. Hint: Like the proof for COF but also let W_e kick the e^{th} marker at most once to make A meet W_e if W_e infinite.

Exercise 40.14. Prove that the family of simple sets is not an r.e. class.

Exercise 40.15. Prove of disprove:

(a) there exists a total $f \leq_T 0^{((2)}$ such that for all e, if W_e is recursive, then $W_{f(e)} = \overline{W_e}$.

(b) there exists a total $f \leq_T 0^{(3)}$ such that for all e, if W_e is recursive, then $W_{f(e)} = \overline{W_e}$.

41 Domination and high degrees

Theorem 41.1 *(Martin) For any set $A \subseteq \omega$ the following are equivalent:*

1. $0'' \leq_T A'$ *and*
2. *there exists $g \leq_T A$ such that $\forall^\infty n \;\; f(n) \leq g(n)$ for every recursive f.*

Proof

(1) $\rightarrow$ (2)

Since TOT is Turing equivalent to $0''$ (40.2), by the limit lemma (28.1) there is a total $h \leq_T A$ so that for every $e \in \omega$

$$\mathrm{T0T}(e) = \lim_{s\to\infty} h(e,s).$$

Define $h_e(x) = h(e,x)$ and define $g(x)$ to be the maximum of the set:

$$\{\{e\}(x) \;:\; e < x \text{ and } \{e\}_s(x)\downarrow \text{ where } h_e\restriction[x,s) \equiv 1\}.$$

Note that if $\{e\}$ is not total, then h_e is eventually zero. If it is total then h_e is eventually one. It is easy to check that $g \leq_T h \leq_T A$ and g eventually dominates each recursive functions.

(2) $\rightarrow$ (1)

Define

$$h(e,s) = \begin{cases} 1 & \text{if } \{e\}_{g(s)}(x)\downarrow \text{ for all } x < s \\ 0 & \text{otherwise.} \end{cases}$$

Then since g eventually dominates all recursive functions we get that

$$\mathrm{T0T}(e) = \lim_{s\to\infty} h(e,s)$$

and hence by the limit lemma that

$$0'' \equiv_T \mathrm{T0T} \leq_T A'.$$

QED

Theorem 41.2 *(Martin, Tennenbaum) If A is a maximal set, then*

$$A'' \equiv_T 0''.$$

Proof
Let $g(n) = \overline{a}_n$ where $\overline{A} = \{\overline{a}_0 < \overline{a}_1 < \cdots\}$. We already know that since A is hypersimple that $\exists^\infty n \quad f(n) < g(n)$ for any recursive f. Suppose $\exists^\infty n \quad g(n) < f(n)$. Then there is a strong array $\langle F_n : n < \omega\rangle$ such that $|F_n \cap \overline{A}| \geq 2$ for infinitely many n. This because there must be infinitely many n with

$$f(n) \leq g(n) < g(n+1) < f(n+1)$$

and hence $F_n = [f(n), f(n+1))$ does the trick. But as in the characterization of hyperhypersimple (35.3 part 4) there is a weak array $\langle H_n \subseteq F_n : n < \omega\rangle$ such that for every n if $F_n \cap \overline{A} \neq \emptyset$, then $|H_n \cap \overline{A}| = 1$. But then

$$A \subseteq A \cup \bigcup_n H_n \subseteq \omega$$

shows that A is not maximal.
QED

Theorem 41.2 is also true for the hyperhypersimple sets. Martin has shown the converse that every high r.e. degree contains a maximal set.

Exercise 41.3. Prove that if for all f partial recursive we have that

$$\forall^\infty n \in dom(f) \;\; f(n) \leq g(n)$$

then $0' \leq_T g$.

Example 41.4 *Suppose A is maximal and $B = A \oplus A$. Then B is not maximal, but $\langle \overline{b}_n : n < \omega\rangle$ eventually dominates every recursive function.*

Proof
It is not maximal since $B \subseteq (B \oplus \text{Evens}) \subseteq \omega$ and these inclusions are non trivial.

To see domination note that $\overline{b}_{2n} = 2\overline{a}_n$ and $\overline{b}_{2n+1} = 2\overline{a}_n + 1$. For any recursive f

$$\forall^\infty n \;\; (f(2n), f(2n+1) < \overline{a}_n < \overline{b}_{2n} < \overline{b}_{2n+1})$$

and hence

$$\forall^\infty m \;\; f(m) < \overline{b}_m.$$

QED

Example 41.5 *There is a r.e. set A which is not hyperhypersimple but $\overline{a}_n$ eventually dominates every recursive function.*

Proof

Let $F_k = [n_k, n_{k+1})$ be the strong array with $n_{k+1} = n_k + k + 1$. Note that $|F_k| = k$. Let B any maximal set. For each k and l if $|B \cap k| = l$ let G_k be the top l elements of F_k. It is easy to see that $\langle G_k \subseteq F_k : k \in \omega\rangle$ is a weak array. Let

$$A = \bigcup G_k \cup \bigcup_{k \in B} F_k.$$

Note that for every k

$$|F_{\overline{b}_k} \setminus G_{\overline{b}_k}| = k.$$

For each $l < \omega$ define

$$P_l = \{n_k + l : l < k < \omega\}$$

then $\langle P_l : l < \omega\rangle$ is a weak array demonstrating that A is not hyperhypersimple. Note that

$$F_{\overline{b}_k} \setminus G_{\overline{b}_k} = \{\overline{a}_i : l_k \leq i < l_{k+1}\}$$

where $l_{k+1} = l_k + k$ and so $l_k = \frac{k(k+1)}{2}$. Hence for any recursive function f we have that

$$\forall^\infty k \;\; f(l_{k+1}) < \overline{b}_k < \overline{a}_{l_k}$$

and so for f increasing we have:

$$\forall^\infty m \;\; f(m) < \overline{a}_m.$$

QED

Exercise 41.6. Prove that for any maximal set B

$$\forall^\infty n \;\; f(\overline{b}_n) < \overline{b}_{n+1}$$

for every recursive f.

42 High degrees using the Psuedojump

Theorem 42.1 *(Shoenfield, Sacks) There is a nontrivial high degree, i.e., there exists a r.e. set A with $A <_T 0'$ and $A'' \equiv_T 0''$.*

Proof
This was originally proved using an infinite injury priority argument. We give here a proof due to Jockusch and Shore which needs only a finite injury priority argument together with relativization and uniformization.

Define that pseudojump operator J_e as follows:

$$J_e(A) = A \oplus W_e^A.$$

Lemma 42.2 *For any e_0 there exists a r.e. set A with*

$$A >_T 0 \text{ and } J_{e_0}(A) \equiv_T 0'.$$

Proof
Let $0' = \{e_s : s < \omega\}$ be a one-to-one recursive enumeration of $0' = K$.

Requirements and strategies

Our requirements can be described as follows:

P_{2e} $\quad \overline{A} \neq W_e$

Our strategy will be to put its follower v_{2e} into A if it ever turns up in W_e.

P_{2e+1} $\quad$ Code e into A if e every turns up in $0'$.

Our strategy will put its follower v_{2e+1} into A_{s+1} if $e = e_s$. We will show that v_{2e+1} can be computed from $J_{e_0}(A)$ and so $0' \leq_T J_{e_0}(A)$.

N_n $\quad (\exists^\infty s \ \{e_0\}_s^{A_s}(n)\downarrow) \rightarrow \{e_0\}^A(n)\downarrow$

This is to insure that $W_{e_0}^A \leq_T 0'$. We use the usual low simple set strategy of restraining the computation.

Construction

For each negative requirement N_n define the restraint function $r(n,s)$ to be the use of the computation $\{e_0\}_s^{A_s}(n)$ (recall that this is zero if the computation does not converge).

For each positive requirement P_n its set of potential followers is

$$F_n = \{\langle n, m\rangle : m < \omega\}$$

and its follower v_n^s at stage s is:

$$v_n^s = \min\{v \in F_n \ : \ v > \max(r(m,s) : m \leq n)\}.$$

We say that P_n requires attention at stage s iff

1. ($n = 2e+1$ and $e = e_s$) or
2. $n = 2e < s$ and $W_e^s \cap A_s = \emptyset$ and $v_n^s \in W_e^s$.

Put $A_{s+1} = A_s \cup \{v_n^s : P_n$ requires attention at stage $s\}$.

Verification

Note that each positive requirement can act at most once. It follows that

$$\lim_{s\to\infty} r(n,s) = r(n) < \infty$$

and each N_n and P_n is met. Hence we have that $W_{e_0}^A \leq_T 0'$ and $A >_T 0$. It remains only to see the following:

Claim. $0' \leq_T A \oplus W_{e_0}^A$.

Define

$$f(n) = \begin{cases} 1 & \text{if } P_n \text{ ever acts} \\ 0 & \text{otherwise.} \end{cases}$$

Obviously $0' \leq_T f$ since $e \in 0'$ iff $f(2e+1) = 1$. So it is enough to see that

$$f \leq_T A \oplus W_{e_0}^A.$$

Assume we have computed $f \restriction n$ using an oracle for $A \oplus W_{e_0}^A$ and we show how to compute $f(n)$. Find a stage s_0 so that

1. $\forall m < n$ if P_m ever acts, it has already acted by stage s_0, and
2. $\forall m \leq n \quad \{e_0\}^A(m)\downarrow$ iff $\{e_0\}_{s_0}^{A_{s_0}}(m)\downarrow$.

Using the oracle ($n \in W^A_{e_0}$?) we can test that s_0 satisfies (2), but since $r(m, s_0)$ will protect the computation $\{e_0\}^{A_{s_0}}_{s_0}(m)$, this will in fact be the correct computation at all stages $s \geq s_0$. It follows that $r(m, s) = r(m, s_0)$ for all $m \leq n$ and $s \geq s_0$. Therefor $v(n, s) = v(n, s_0)$ for all $s \geq s_0$. Hence P_n will act iff either it has already by stage s_0 or $v(n, s_0) \in A$ (which happens iff it acts after s_0).

This proves the Claim and the Lemma.

QED

Next we need to see that a relativized and uniformitized version of the Lemma is true.

The Lemma says:

For all e there exists a r.e. set A such that $A >_T 0$ and $J_e(A) \equiv 0'$.

The uniformized version says:

There exists a recursive $f : \omega \to \omega$ such that for all e

$$W_{f(e)} >_T 0 \text{ and } J_e(W_{f(e)}) \equiv 0'.$$

Or using the psuedojump we could equivalently prove:

There exists a recursive $f : \omega \to \omega$ such that for all e

$$J_{f(e)}(0) >_T 0 \text{ and } J_e(J_{f(e)}(0)) \equiv 0'.$$

The same proof will work for every oracle B. So finally we get the relativized and uniformitized version:

Lemma 42.3 *There exists a recursive $f : \omega \to \omega$ such that for all e and for all $B \subseteq \omega$:*

$$J_{f(e)}(B) >_T B \text{ and } J_e(J_{f(e)}(B)) \equiv B'.$$

Fix f from Lemma 42.3 and consider any $n > 0$ and e.

Proposition 42.4 *Suppose*

$$\forall B \ \ (J_e(B))^{(n)} \equiv_T B^{(n)} \text{ and } (J_e(B))^{(n-1)} \not\equiv_T B^{(n-1)}$$

Then

$$\forall B \ \ (J_{f(e)}(B))^{(n)} \equiv_T B^{(n+1)} \text{ and } (J_{f(e)}(B))^{(n-1)} \not\equiv_T B^{(n)}.$$

Proof
Note that
$$(J_{f(e)}(B))^{(n)} \equiv_T (J_e(J_{f(e)}(B)))^{(n)}$$
by substituting $J_{f(e)}(B)$ for B in the hypothesis. We also have that
$$(J_e(J_{f(e)}(B)))^{(n)} \equiv_T B^{(n+1)}$$
because $J_e(J_{f(e)}(B)) \equiv_T B'$ and hence
$$(J_{f(e)}(B))^{(n)} \equiv_T B^{(n+1)}.$$
Similarly by substituting $J_{f(e)}(B)$ for B in the hypothesis
$$(J_{f(e)}(B))^{(n-1)} \not\equiv_T (J_e(J_{f(e)}(B)))^{(n-1)} \equiv_T B^{(n)}$$
and so
$$(J_{f(e)}(B))^{(n-1)} \not\equiv_T B^{(n)}.$$
QED

We are using the terminology $B^{(0)} = B$.

By a similar proof we have

Proposition 42.5 *Suppose*
$$\forall B \;\; (J_e(B))^{(n)} \equiv_T B^{(n+1)} \text{ and } (J_e(B))^{(n-1)} \not\equiv_T B^{(n)}.$$
Then
$$\forall B \;\; (J_{f(e)}(B))^{(n+1)} \equiv_T B^{(n+1)} \text{ and } (J_{f(e)}(B))^{(n)} \not\equiv_T B^{(n)}.$$

Define the high low hierarchy of r.e. degrees as follows

1. $H_0 = \{o'\}$
2. $L_0 = \{o\}$
3. $L_n = \{a \in \mathcal{R} \;:\; a^{(n)} = o^{(n)}\}$
4. $H_n = \{a \in \mathcal{R} \;:\; a^{(n)} = o^{(n+1)}\}$

Choose e_0 so that for every B, $B' \equiv_T J_{e_0}(B)$. Let $e_{n+1} = f(e_n)$. And let $\mathbf{a}_n$ be the Turing degree of $J_{e_n}(0)$.

Proposition 42.6 *For every n*

$$\mathbf{a}_{2n} \in H_n \setminus H_{n-1} \text{ and } \mathbf{a}_{2n+1} \in L_{n+1} \setminus L_n.$$

Proof

Note that

$$\forall B \;\; (J_{f(e_0)}(B))^{(1)} \equiv J_{e_0}(J_{f(e_0)}(B)) \equiv B^{(1)}$$

but

$$(J_{f(e_0)}(B))^{(0)} \not\equiv B^{(0)}.$$

So applying the Propositions 42.4 and 42.5 alternatingly, the result follows. QED

Note that in particular, $\mathbf{a}_2$ is a nontrivial high degree and this proves Theorem 42.1.

Theorem 42.7 *(Martin, Lachlan, Sacks) There exist a r.e. degree* $\mathbf{a}$ *such that*

$$\mathbf{a} \notin \bigcup_{n<\omega}(L_n \cup H_n).$$

Proof

In Lemma 42.3 take e_0 to be fixed point for f and hence $J_{e_0}(B) = J_{f(e_0)}(B)$ for every set B. Define $H(B) = J_{e_0}(B)$ where H is short for the Hop of B. Then for every A we have

$$B <_T H(B) <_T H^2(B) \equiv_T B'$$

i.e., two hops make a jump. Hence if $A = H(0)$, then for every n we have that $A^{(n)} = H^{2n+1}(0)$ and so

$$0^{(n)} \equiv_T H^{2n}(0) < TH^{2n+1}(0) \equiv_T A^{(n)} <_T H^{2n+2}(0) \equiv_T 0^{(n+1)}.$$

QED

M.Simpson found a proof of the Sack's Jump Theorem using the psuedo-jump. It appears in Soare.

Exercise 42.8. Prove or disprove: For any e if $A \leq_T B$ then $W_e^A \leq_T W_e^B$.

43 First-order theories

In this section we give two examples of first-order theories with interesting properties. All theories in this section are assumed to be in a recursively presented language. Craig noted that being axiomatized by an r.e. set of sentences is equivalent to having a recursive set of axioms. Given an r.e. list $\theta_0, \theta_1, \ldots,$ replace it by the recursive list:

$$\theta_0, (\theta_0 \wedge \theta_1), \ldots, (\theta_0 \wedge \theta_1 \wedge \cdots \wedge \theta_n), \ldots$$

Lemma 43.1 *(Shoenfield) There exist an r.e. set B such that*

1. $B <_T 0'$,
2. $\forall e \in TOT \quad B_e =^* \omega$,
3. $\forall e \notin TOT \quad B_e =^* \emptyset$, *and*
4. $\forall e, n \quad n \in B_e \to \{e\}(n) \downarrow$.

Proof

Recall that $e \in TOT$ iff $\{e\}$ is a total function.

By Theorem 42.1 there exists an r.e. set $A <_T 0'$ with $A' \equiv_T 0''$ and hence by Theorem 41.1 there exists $g \leq_T A$ such that for every recursive f we have $\forall^\infty n \;\; f(n) \leq g(n)$. Let $A = \bigcup_s A_s$ be a recursive enumeration of A and suppose $g = \{e_0\}^A$.

Using a permitting argument we will get $B \leq_T A$. Put

$$B_{s+1} = B_s \cup \{\langle e, n\rangle < s \;:\; \{e_0\}_s^{A_s}(n) \downarrow = t \text{ and } \forall m \leq n \;\; \{e\}_t(m) \downarrow\}.$$

It is easy to check that B has properties (2),(3), and (4).

We show that $B \leq_T A$. To decide whether $\langle e, n\rangle \in B$ find a stage s_0 such that $\{e_0\}_s^{A_s}(n) \downarrow$ with use u and $A_{s_0} \cap [0, u] = A \cap [0, u]$. But this means that $\langle e, n\rangle \in B$ iff $\langle e, n\rangle \in B_{s_0+1}$.

QED

Definition 43.2 *For a first-order theory T in a language containing a sequence of terms $\underline{n}$ for $n < \omega$ we say that*

1. *$R \subseteq \omega$ is weakly represented in T iff there is a formula $\theta(x)$ such that*

$$\forall n \;\; (n \in R \textit{ iff } T \vdash \theta(\underline{n})).$$

2. *$R \subseteq \omega$ is strongly represented in T iff there is a formula $\theta(x)$ such that*

$$\forall n \;\; (n \in R \to T \vdash \theta(\underline{n}))$$

and

$$\forall n \;\; (n \notin R \to T \vdash \neg\theta(\underline{n})).$$

Proposition 43.3 *Assume T is recursively axiomatizable.*

1. *Strongly representable implies weakly representable.*
2. *Weakly representable sets are recursively enumerable.*
3. *Strongly representable sets are recursive.*
4. *If every recursive set is weakly represented in T, then T is undecidable.*
5. *If every r.e. set is weakly representable in T, then the decision problem for T is equivalent to $0'$.*

Proof

(4)

If T is decidable, then there is a recursive predicate $U \subseteq \omega \times \omega$ which is universal for all $R \subseteq \omega$ which are weakly represented in T. But then the recursive set $D = \{n \;:\; \langle n, n\rangle \notin U\}$ cannot be weakly represented in T.

(5)

By the decision problem for T we mean the Turing degree of the set:

$$\{\theta \;:\; T \vdash \theta\}.$$

This result is clear since $0'$ is weakly represented in T.
QED

Example 43.4 *(Shoenfield) There is a recursively axiomatizable theory T in which every recursive set is strongly represented but the decision problem for T is of degree strictly smaller than $0'$.*

Proof

The language of T consists of infinitely many constant symbols $\underline{n}$ and unary predicate symbols R_n for $n < \omega$. Let B be the set from Lemma 43.1. The axioms of T are the following:

1. $\underline{n} \neq \underline{m}$ for $n < m < \omega$,
2. $R_e(\underline{m})$ if $\langle e, m\rangle \in B$ and $\{e\}(m) \downarrow= 1$,
3. $\neg R_e(\underline{m})$ if $\langle e, m\rangle \in B$ and $\{e\}(m) \downarrow= 0$, and
4. infinitely many axioms saying the predicates R_e are independent. i.e., for each pair of disjoint finite sets $G, H \subseteq \omega$:

$$\exists v \ (\bigwedge_{e\in G} R_e(v) \ \ \wedge \ \ \bigwedge_{e\in H} \neg R_e(v)).$$

This is an r.e. set but by Craig's trick T is recursively axiomatizable.

Every recursive set is strongly represented in T. If $R \subseteq \omega$ is recursive, then for some e we have that $\{e\}$ is the characteristic function of R. Since $B_e =^* \omega$ we know that the formula $R_e(v)$ almost represents R. It is easy to tweak it to represent R.

The decision problem for T is Turing reducible to B. This follows from the fact that T eliminates quantifiers.
QED

Definition 43.5 *A set of sentences Σ is independent iff $\Sigma \not\vdash \theta$ for any $\theta \in \Sigma$.*

Tarski proved that any first-order theory in a countable language is independently axiomatizable. Reznikoff proved it for uncountable languages.

Lemma 43.6 *Suppose a theory T can be axiomatized by an infinite recursively enumerable independent set Σ. Then for any r.e. set of sentences $\langle \rho_n : n < \omega\rangle$ axiomatizing T there is a recursive function f such that for every $n > 0$:*

$$\bigwedge_{k<n} \rho_k \text{ does not imply } \bigwedge_{k<f(n)} \rho_k.$$

Example 43.7 *(Kreisel) There is a recursively axiomatizable theory T which cannot be axiomatized by a recursively enumerable independent set of sentences.*

Proof

Robinson's theory Q is a finite subset of Peano Arithmetic PA which is in turn a subtheory of true arithmetic $(\omega, +, \cdot, S, 0)$, i.e.,

$$Q \subseteq PA \subseteq Th(\omega, +, \cdot, S, 0).$$

Every r.e. set is weakly represented in Q. Let $H \subseteq \omega$ be any hypersimple set and suppose $\theta(v)$ is a formula such that

$$\forall n \quad (n \in H \text{ iff } Q \vdash \theta(\underline{n})).$$

Let T be the theory in the language of arithmetic plus one new unary predicate symbol R with the following set of axioms:

1. $\bigwedge Q$,
2. $\forall v \ \ (\theta(v) \to R(v))$, and
3. infinitely many axioms:
$$R(\underline{n})$$
for each $n < \omega$.

The symbol $\underline{n}$ is shorthand for $S(S(\cdots S(0))\cdots)$ with n many S's. Let f be the recursive function from Lemma 43.6. Since H is hypersimple there exists n such that $[n, f(n)) \subseteq H$. But then for all k in $[n, f(n))$

$$Q \vdash \theta(\underline{n}).$$

Hence

$$[\bigwedge Q \quad \wedge \quad \forall v \ \ \theta(v) \to R(v)] \quad \vdash \quad \bigwedge_{n \leq k < f(n)} R(\underline{n})$$

which is a contradiction.
QED

44 Analytic sets

Definition 44.1 *$A \subseteq \omega^\omega$ is Σ^1_1 iff there exists a recursive $R \subseteq \omega^{<\omega} \times \omega^{<\omega}$ such that*

$$x \in A \equiv \exists y \in \omega^\omega \ \ \forall n \in \omega \ \ R(x\restriction n, y\restriction n).$$

Similarly $A \subseteq \omega$ is Σ^1_1 iff there exists a recursive $R \subseteq \omega \times \omega^{<\omega}$ such that

$$k \in A \equiv \exists y \in \omega^\omega \ \ \forall n \in \omega \ \ R(k, y\restriction n).$$

Π^1_1 sets are the complements of Σ^1_1 sets and $\Delta^1_1 = \Pi^1_1 \cap \Sigma^1_1$.

We can give similar definitions of Σ^1_1 and Σ^0_n and Π^0_n for $\mathcal{X}$ any finite product $\mathcal{X} = \prod_{i<N} X_i$ where each X_i is either ω or ω^ω.

Proposition 44.2 *1.* $\Pi^0_1 \subseteq \Sigma^1_1$

2. If $A \subseteq \mathcal{X} \times \omega^\omega$ *is* Σ^1_1 *then* B *is* Σ^1_1 *where*

$$B(x) \text{ iff } \exists y \ \ A(x,y)$$

3. If A *and* B *are* Σ^1_1 *then* $A \wedge B$ *and* $A \vee B$ *are* Σ^1_1.

4. If $A \subseteq \omega \times \mathcal{X}$ *is* Σ^1_1 *then both*

(a) $B(x) \equiv \exists n \in \omega \ \ A(n,x)$ *and*

(b) $C(x) \equiv \forall n \in \omega \ \ A(n,x)$

are Σ^1_1.

Proof

(1) trivial

(2) Suppose $\mathcal{X} = \omega^\omega$ and

$$A(x,y) \equiv \exists z \ \ \forall n \ \ R(x \restriction n, y \restriction n, z \restriction n)$$

define

$$R^*(\sigma, \tau) \text{ iff } R(\sigma, \tau_0, \tau_1) \text{ where } \tau(i) = \langle \tau_0(i), \tau_1(i) \rangle$$

Then

$$B(x) \equiv \exists u \ \ \forall n \ \ R^*(x \restriction n, u \restriction n)$$

(3) Suppose

$$A(x) \equiv \exists y \ \ C(x,y)$$
$$B(x) \equiv \exists z \ \ D(x,z)$$

where C and D are Π^0_1. Then

$$A(x) \vee B(x) \ \ \equiv \ \ \exists w \ \ (C(x,w) \vee D(x,w))$$

and

$$A(x) \wedge B(x) \ \ \equiv \ \ \exists y \, \exists z \ \ (C(x,y) \wedge D(x,z))$$

(4a) Suppose

$$A(n,x) \equiv \exists y\ \ \forall m\ \ R(n, x\restriction m, y\restriction m)$$

Define

$$R^*(x\restriction n, y\restriction n) \text{ iff } R(y(0), x\restriction(n-1), y^*\restriction(n-1)) \text{ where } y^*(i) = y(i+1)$$

Then

$$B(x) \equiv \exists n\ A(n,x) \equiv \exists y\ \ \forall m\ \ R^*(x\restriction m, y\restriction m)$$

(4b) Suppose

$$A(n,x) \equiv \exists y\ \ \forall m\ \ R(n, x\restriction m, y\restriction m)$$

Define
$R^*(x\restriction m, z\restriction m)$ iff $R(i, x\restriction j, y_i\restriction j)$ for each $\langle i,j\rangle < m$ and $y_i(j) = z(\langle i,j\rangle)$.
Then

$$C(x) \equiv \forall n\ A(n,x) \equiv \exists z\ \ \forall m\ \ R^*(x\restriction m, z\restriction m)$$

QED

Proposition 44.3 *Universal Σ^1_1 sets exists, hence $\Sigma^1_1 \neq \Pi^1_1$.*

Proof
Let $U \subseteq \omega \times \mathcal{X} \times \omega^\omega$ be a universal Π^0_1 set for subsets of $\mathcal{X} \times \omega^\omega$, then

$$V(n,x) \equiv \exists y\ \ A(n,x,y)$$

is Universal Σ^1_1.
QED

Theorem 44.4 *(Tennenbaum) There exists a recursive linear order $(\omega, \trianglelefteq)$ which is isomorphic to $\omega + \omega^*$ with the property that every nonempty r.e. subset of ω has a $\trianglelefteq$-least and $\trianglelefteq$-greatest element.*

Proof
Note that ω^* stands for reverse ω or equivalently the order type of the negative integers. Let

$$L = \{x \in \omega\ :\ |\{y : y \vartriangleleft x\}| < \omega\} \text{ and } R = \{x \in \omega\ :\ |\{y : x \vartriangleleft y\}| < \omega\}$$

In our construction we make sure that $\omega = L \cup R$ and each is infinite. At stage s we assume that we have (effectively) determined the finite linear order $\trianglelefteq\upharpoonright(s \times s)$ and just decide where to put the new element, s, of

$$s + 1 = \{0, 1, 2, \ldots, s\}.$$

Our requirements are:

R_e $\quad W_e$ infinite $\to W_e \cap L \neq \emptyset$ and $W_e \cap R \neq \emptyset$.

We assume at stage s in our construction that some requirements R_e, say $e \in F_s \subseteq s$, have followers $l_e < s$ and $r_e < s$ which satisfy:
if $e < e'$ and $e, e' \in F_s$, then $l_e \lhd l_{e'} \lhd r_{e'} \lhd r_e$.
At stage $s + 1$ we look for the smallest $e < s$ (if any) such that

1. $e \notin F_s$ (or equivalently R_e has no followers)
2. there exists $l, r \in W_{e,s}$ such that for every $e' < e$ with $e' \in F_s$ we have that

$$l_{e'} \lhd l \lhd r \lhd r_{e'}$$

For the smallest such e and smallest such pair l, r we appoint $l = l_e$ and $r = r_e$ the followers of R_e and put

$$F_{s+1} = \{e' < e \ : \ e' \in F_s\} \cup \{e\}$$

i.e., we unappoint all followers for $e' > e$. If there is no such e we do not change any followers.

In either case, we put s into the ordering $\trianglelefteq\upharpoonright(s \times s)$ in the first gap above all the l_e for $e \in F_{s+1}$ (and therefore, below all the r_e for $e \in F_{s+1}$.)

Claim. For each e if W_e is infinite, then R_e obtains permanent followers l_e and r_e and is met.
Proof
Suppose the Claim is true for all $e' < e$. Suppose s_0 is a large enough stage so that no $e' < e$ acts after stage s_0. Let e_0 be the maximum element of F_{s_0} below e. Then since $s > s_0$ are put between l_{e_0} and r_{e_0} and W_e is infinite, it must be that some followers are appointed to R_e if it doesn't already have them. These followers are permanent.
QED

Since infinitely many W_e are infinite and hence acquire permanent followers, it must be that L and R are infinite and therefore the order type we construct is $\omega + \omega^*$.
QED

Corollary 44.5 *(Jockusch) There exists a recursive function $f : [\omega]^2 \to 2$ such that there is no infinite recursive $H \in [\omega]^\omega$ such that $f \restriction [H]^2$ is constant.*

Proof
Define

$$f(x,y) = \begin{cases} 1 & \text{if } x < y \to x \lhd y \\ 0 & \text{if } x < y \to y \lhd x \end{cases}$$

QED

Definition 44.6 *$T \subseteq \omega^{<\omega}$ is a well-founded tree iff*
(a) $\forall \sigma, \tau \;\; \sigma \subseteq \tau \in T \to \sigma \in T$
(b) T has no infinite branch, i.e., $[T] = \emptyset$ where

$$[T] =^{def} \{x \in \omega^\omega \; : \; \forall n \;\; x \restriction n \in T\}.$$

Definition 44.7 *(Kleene-Brouwer ordering) For $\sigma, \tau \in \omega^{<\omega}$*

$$\sigma <_{KB} \tau \textit{ iff } \sigma \supsetneq \tau \textit{ or } \exists n < \min(|\sigma|, |\tau|) \;\; \sigma \restriction n = \tau \restriction n \textit{ and } \sigma(n) < \tau(n)$$

$$\sigma \leq_{KB} \tau \textit{ iff } \sigma <_{KB} \tau \textit{ or } \sigma = \tau$$

Proposition 44.8 *$\leq_{KB}$ is a recursive linear ordering of $\omega^{<\omega}$.*

Theorem 44.9 *(Kleene-Brouwer) Given a tree $T \subseteq \omega^{<\omega}$*
T is well-founded iff $(T, \leq_{KB})$ is a well-ordering.

Proof
Suppose that T is not well-founded and $x \in [T]$. Then for each n

$$x \restriction (n+1) <_{KB} x \restriction n$$

and so $(T, \leq_{KB})$ is not a well-ordering.

Conversely, suppose that $(T, \leq_{KB})$ is not a well-ordering and $(\sigma_n \in T \; : \; n < \omega)$ is $<_{KB}$-descending, i.e.,

$$\sigma_{n+1} <_{KB} \sigma_n.$$

Then an easy induction produces $x \in \omega^\omega$ with the property that

$$\forall n \ \forall^\infty m \quad x \restriction n \subseteq \sigma_m.$$

It follows that $x \in [T]$ and so T is not well-founded.
QED

Definition 44.10 *For $T \subseteq \omega^{<\omega}$ a tree and α an ordinal we define $T_\alpha \subseteq T$ as follows:*
(a) $\sigma \in T_0$ iff $\sigma \in T$ and $\forall n \ \ \sigma n \notin T$. (Terminal nodes of T.)
(b) $\sigma \in T_\alpha$ iff $\sigma \in T$ and $\forall n \ \ (\sigma n \in T \to \sigma n \in T_{<\alpha})$.
(c) $T_{<\alpha} =^{def} \cup_{\beta<\alpha} T_\beta$.

Definition 44.11 *For $\sigma \in T$*
(a) $rank_T(\sigma) = \alpha$ where α is the smallest ordinal with $\sigma \in T_\alpha$.
(b) $rank_T(\sigma) = \infty$ if there is no such α.

Proposition 44.12 *For $T \subseteq \omega^{<\omega}$ a tree, T is well-founded iff $rank_T(\langle\rangle) < \infty$, i.e., its an ordinal.*

Proof
Note that if $rank_T(\sigma) = \infty$, then there exists n such that $rank_T(\sigma n) = \infty$. Hence, $rank_T(\langle\rangle) = \infty$ implies that T has an infinite branch. On the other hand if $rank_T(\sigma) < \infty$, then for every n with $\sigma n \in T$ we have that

$$rank_T(\sigma n) < rank_T(\sigma)$$

Hence T cannot have an infinite branch.
QED

Definition 44.13 *$c : T \to \omega$ is a hypcode iff $T \subseteq \omega^{<\omega}$ is a recursive well-founded tree and c is partial recursive map with domain T. Given a hypecode c we define the sets $H(c, \sigma)$ as follows by induction on the rank of σ. Fix $U \subseteq \omega \times \mathcal{X}$ a universal Σ^0_1 set.*

(a) for $\sigma \in T_0$ a terminal node of T

$$H(c, \sigma) = U_{c(\sigma)}$$

(b) for $\sigma \in T$ not terminal and $c(\sigma) = 0$

$$H(c, \sigma) = \cup_{n, \sigma n \in T} H(c, \sigma n)$$

(c) for $\sigma \in T$ not terminal and $c(\sigma) > 0$

$$H(c, \sigma) = \cap_{n, \sigma n \in T} H(c, \sigma n)$$

$A \subseteq \mathcal{X}$ is hyperarithmetic (HYP) iff there exists a hypcode c and

$$A = H(c) =^{def} H(c, \langle\rangle).$$

Proposition 44.14 *HYP $\subseteq \Delta^1_1$.*

Proof
$x \in H(c)$ iff there exists $f : T \to \{0, 1\}$ such that

1. $\forall \sigma \in T_0$
$$f(\sigma) = 1 \text{ iff } x \in U_{c(\sigma)}$$

2. $\forall \sigma \in T \setminus T_0$ if $c(\sigma) = 0$ then
$$f(\sigma) = 1 \text{ iff } \exists n \ \ (\sigma n \in T \quad \wedge \quad f(\sigma n) = 1)$$

3. $\forall \sigma \in T \setminus T_0$ if $c(\sigma) > 0$ then
$$f(\sigma) = 1 \text{ iff } \forall n \ \ (\sigma n \in T \to f(\sigma n) = 1)$$

4. $f(\langle\rangle) = 1$

It is easy to check that $1 - 4$ are all arithmetic predicates and so $H(c)$ is Σ^1_1.
To see that the complement of $H(c)$ is also Σ^1_1 just note that
$x \notin H(c)$ iff there exists $f : T \to \{0, 1\}$ such that
1,2,3, and
$4'$. $f(\langle\rangle) = 0$.
QED

Theorem 44.15 *(Kleene-Souslin)*
Suppose A and B are disjoint Σ^1_1 sets. Then they can be separated by a hyperarithmetic set C. Hence HYP $= \Delta^1_1$.

Proof
To simplify the notation we assume that $A, B \subseteq \omega^\omega$ although essentially the same proof will work for $A, B \subseteq \omega$ or any $\mathcal{X}$. Since A, B are Σ^1_1 there are recursive trees

$$T^A, T^B \subseteq \cup_{n<\omega}\omega^n \times \omega^n$$

such that

$$x \in A \text{ iff } \exists y \;\; \forall n \;\; (x\restriction n, y\restriction n) \in T^A$$

$$x \in B \text{ iff } \exists z \;\; \forall n \;\; (x\restriction n, z\restriction n) \in T^B$$

The fact that A and B are disjoint implies that it is impossible to find (x, y, z) such that $(x\restriction n, y\restriction n) \in T^A$ and $(x\restriction n, z\restriction n) \in T^B$ for all n. This tells us how to find our recursive well-founded tree T.

Given $\rho \in \omega^{<\omega}$ we determine a triple $trip(\rho) = (\sigma, \tau_1, \tau_2)$ by the rule that $\sigma(i) = \rho(3i)$, $\tau_1(i) = \rho(3i+1)$, and $\tau_2(i) = \rho(3i+2)$. We take the natural length functions, namely

- $|\sigma| = |\tau_1| = |\tau_2| = n$ if $|\rho| = 3n$,
- $|\sigma| = n+1, \;\; |\tau_1| = |\tau_2| = n$ if $|\rho| = 3n+1$, and
- $|\sigma| = |\tau_1| = n+1, \;\; |\tau_2| = n$ if $|\rho| = 3n+2$.

Now we define the recursive well-founded tree $T \subseteq \omega^{<\omega}$ and hypcode $c : T \to \omega$ as follows:

1. for $\rho \in \omega^{<\omega}$ with length $|\rho| = 3n+2$ and $trip(\rho) = (\sigma, \tau_1, \tau_2)$ if

   (a) $(\sigma\restriction n, \tau_1\restriction n) \in T^A$,

   (b) $(\sigma\restriction n, \tau_2) \in T^B$, and

   (c) $(\sigma, \tau_1) \notin T^A$,

   then ρ is a terminal node of T and put $c(\rho) = n_0$ where

   $$U_{n_0} = \emptyset.$$

2. for $\rho \in \omega^{<\omega}$ with length $|\rho| = 3(n+1)$ and $trip(\rho) = (\sigma, \tau_1, \tau_2)$ if

   (a) $(\sigma, \tau_1) \in T^A$,

   (b) $(\sigma\restriction n, \tau_2\restriction n) \in T^B$, and

(c) $(\sigma, \tau_2) \notin T^B$,

then ρ is a terminal node of T and put $c(\rho) = n_1$ where

$$U_{n_1} = [\sigma] =^{def} \{x \in \omega^\omega \ : \ \sigma \subseteq x\}.$$

3. For any other ρ we put ρ into T iff it is a proper subset of a terminal node of T. For these ρ we put $c(\rho) = 0$ if $|\rho| = 3n$ or $|\rho| = 3n + 1$ and put $c(\rho) = 1$ if $|\rho| = 3n + 2$.

Now given $trip(\rho) = (\sigma, \tau_1, \tau_2)$ define the following sets:

$$A_\rho = \{x \in [\sigma] \ : \ \exists y \supseteq \tau_1 \ \ \forall n \ \ (x{\restriction}n, y{\restriction}n) \in T^A\}$$

$$B_\rho = \{x \in [\sigma] \ : \ \exists z \supseteq \tau_2 \ \ \forall n \ \ (x{\restriction}n, z{\restriction}n) \in T^B\}$$

To finish the proof we verify the following:

Claim. For each $\rho \in T$ let $trip(\rho) = (\sigma, \tau_1, \tau_2)$ then

$$A_\rho \subseteq H(c, \rho) \subseteq [\sigma]$$

and

$$B_\rho \subseteq [\sigma] \setminus H(c, \rho)$$

Proof

Case ρ a terminal node of T.
Note that in case 1 of the definition of T, we have that A_ρ is the empty set and $c(\sigma)$ is a code for the empty set and so its OK. In case 2 of the definition of T, we have that B_ρ is the empty set and $c(\sigma)$ is a code for $[\sigma]$ and so its OK.

Case $|\rho| = 3n$ and ρ not terminal.
Note that for nonterminal nodes ρ we have that for every k that $\rho k \in T$. In this case $trip(\rho k) = (\sigma k, \tau_1, \tau_2)$.

$$A_{\rho k} = [\sigma k] \cap A_\rho$$

$$B_{\rho k} = [\sigma k] \cap B_\rho$$

and by induction

$$A_\rho = \cup_{k<\omega} A_{\rho k} \subseteq \cup_{k<\omega} H(c, \rho k) =^{def} H(c, \rho) \subseteq [\sigma]$$

($c(\rho) = 0$, so we take unions)

$$B_\rho = \cup_{k<\omega} B_{\rho k} \subseteq \cup_{k<\omega}([\sigma k] \setminus H(c, \rho k)) = [\sigma] \setminus H(c, \rho)$$

The last equality holds because each $H(c, \rho k) \subseteq [\sigma k]$ and $([\sigma k] : k < \omega)$ is a partition of $[\sigma]$.

Case $|\rho| = 3n + 1$ and ρ not terminal.
In this case $trip(\rho k) = (\sigma, \tau_1 k, \tau_2)$, and also $c(\rho) = 0$, i.e., we take unions. Note that for every k that $B_{\rho k} = B_\rho$ since neither σ nor τ_2 change. Also, by the definition of A_ρ note that

$$A_\rho = \cup_{k<\omega} A_{\rho k}.$$

Now by inductive hypothesis we have that

$$A_\rho = \cup_{k<\omega} A_{\rho k} \subseteq \cup_{k<\omega} H(c, \rho k) =^{def} H(c, \rho)$$

$$B_\rho \subseteq [\sigma] \setminus H(c, \rho k)$$

for every k so

$$B_\rho \subseteq [\sigma] \setminus H(c, \rho)$$

as was to be proved.

Case $|\rho| = 3n + 2$ and ρ not terminal.
In this case $trip(\rho k) = (\sigma, \tau_1, \tau_2 k)$, and $c(\rho) = 1$, i.e., take intersections. Note that for every k that $A_{\rho k} = A_\rho$ since neither σ nor τ_1 change. Now by inductive hypothesis we have that

$$A_\rho \subseteq \cap_{k<\omega} H(c, \rho k) =^{def} H(c, \rho)$$

$$B_\rho = \cup_{k<\omega} B_{\rho k} \subseteq \cup_{k<\omega} [\sigma] \setminus H(c, \rho k) = [\sigma] \setminus H(c, \rho)$$

as was to be proved.

This proves the Claim. However since $A_{\langle\rangle} = A$ and $B_{\langle\rangle} = B$ the Theorem follows.
QED

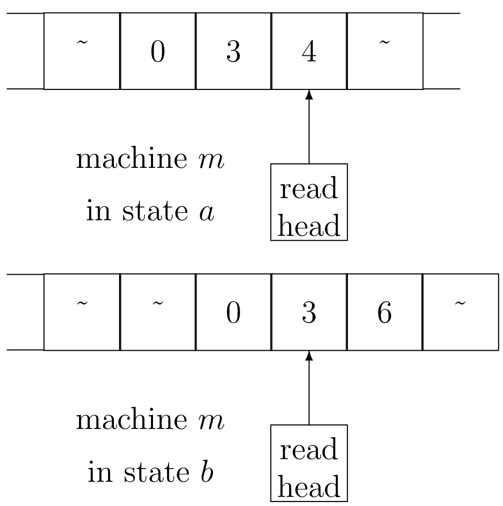
Appendix

45 Turing machines

In this section we define the notion of Turing computable function and include Turing's analysis of why every effectively calculable function should be Turing computable. We also sketch the proof of a universal Turing machine.

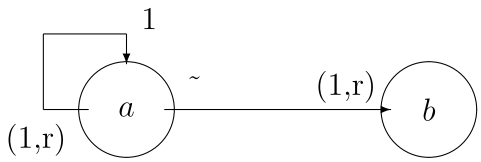
A *Turing machine* is a function m such that for some finite sets A and S the domain of m is a subset of $S \times A$ and range of m is a subset of $S \times A \times \{l, r\}$. We call A the alphabet and S the states.[1]

For example, suppose S is the set of letters $\{a, b, c, \ldots, z\}$ and A is the set of all integers less than seventeen, then $m(a, 4) = (b, 6, l)$ would mean that when the machine m is in state a reading the symbol 4 it will go into state b, erase the symbol 4 and write the symbol 6 on the tape square where 4 was, and then move left one square.



If $(a, 4)$ is not in the domain of m, then the machine halts. This is the only way of stopping a calculation. Let $A^{<\omega}$ be the set of all finite strings from the alphabet A.

[1]This section is taken from my book:
http://www.math.wisc.edu/~miller/res/index.html
see
Introduction to Mathematical Logic - Moore style



$$m(a,1) = (a,1,r)$$
$$m(a,\tilde{\ }) = (b,1,r)$$

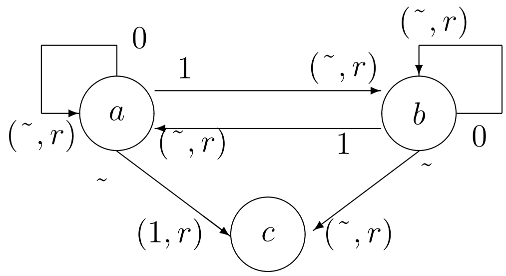
Figure 7: Successor function

The Turing machine m gives rise to a partial function M from $A^{<\omega}$ to $A^{<\omega}$ as follows. We suppose that A always contains the blank space symbol ˜ ; and S contains the starting state a. Given any word w from $A^{<\omega}$ we imagine a tape with w written on it and blank symbols everywhere else. We start the machine in state a and reading the leftmost symbol of w. A configuration consists of what is written on the tape, which square of tape is being read, and the state the machine is in. Successive configurations are obtained according to rules determined by m, namely if the machine is in state q reading symbol s and $m(q,s) = (q', s', d)$ then the next configuration has the same tape except the square we were reading now has the symbol s' on it, the new state is q', and the square being read is one to the left if $d = l$ and one to the right if $d = r$. If (q,s) is not in the domain of m, then the computation halts and $M(w) = v$ where v is what is written on the tape when the machine halts.

Suppose B is a finite alphabet that does not contain the blank space symbol ˜ then a function $f : B^{<\omega} \to B^{<\omega}$ is a *partial Turing computable function* iff there is a Turing machine m with an alphabet $A \supseteq B$ such that $f = M \restriction B^{<\omega}$. A partial Turing computable function is *Turing computable* iff it is total. A function $f : \omega \to \omega$ is Turing computable if it is Turing computable when considered as a map from $B^{<\omega}$ to $B^{<\omega}$ where $B = \{1\}$. Words in B can be regarded as numbers written in base one, hence we identify the number x with x ones written on the tape.

For example, the identity function is Turing computable, since it is computed by the empty machine. The successor function is Turing computable since it is computed by the machine in Figure 7.

In the diagram on the left, states are represented by little circles. The arrows represent the *state transition function* m. For example, the horizontal arrow represents the fact that when m is in state a and reads ˜ then it writes 1, moves right, and goes into state b.



$$
\begin{aligned}
m(a,0) &= (a, \tilde{\ }, r) \\
m(a,1) &= (b, \tilde{\ }, r) \\
m(b,0) &= (b, \tilde{\ }, r) \\
m(b,1) &= (a, \tilde{\ }, r) \\
m(a, \tilde{\ }) &= (c, 1, r) \\
m(b, \tilde{\ }) &= (c, \tilde{\ }, r)
\end{aligned}
$$

Figure 8: Parity checker

The set of strings of zeros and ones with an even number of ones is Turing computable. Its characteristic function (parity checker) can be computed by the machine in Figure 8.

The following problems are concerned with Turing computable functions and predicates on ω.

Exercise 45.1. Show that any constant function is Turing computable.

Exercise 45.2. A binary function $f : \omega \times \omega \to \omega$ is Turing computable iff there is a machine such that for any $x, y \in \omega$ inputing x ones and y ones separated by "," the machine eventually halts with $f(x, y)$ ones on the tape. Show that $f(x, y) = x + y$ is Turing computable.

Exercise 45.3. Show that $g(x, y) = xy$ is Turing computable.

Exercise 45.4. Let $x \dot{-} y = max\{0, x - y\}$. Show that $p(x) = x \dot{-} 1$ is Turing computable. Show that $q(x, y) = x \dot{-} y$ is Turing computable.

Exercise 45.5. Suppose $f(x)$ and $g(x)$ are Turing computable. Show that $f(g(x))$ is Turing computable.

Exercise 45.6. Formalize a notion of multitape Turing machine. Show that we get the same set of Turing computable functions.

Exercise 45.7. Show that we get the same set of Turing computable functions even if we restrict our notion of computation to allow only tapes that are infinite in one direction.

Exercise 45.8. Show that the family of Turing computable functions is closed under arbitrary compositions, for example $f(g(x,y),h(x,z),z)$. More generally, if $f(y_1,\ldots,y_m)$, $g_1(x_1,\ldots,x_n),\ldots$, and $g_m(x_1,\ldots,x_n)$ are all Turing computable, then so is

$$f(g_1(x_1,\ldots,x_n),\ldots,g_m(x_1,\ldots,x_n)).$$

Exercise 45.9. A set is Turing computable iff its characteristic function is. Show that the binary relation $x = y$ is Turing computable. Show that the binary relation $x \leq y$ is Turing computable.

Exercise 45.10. Define

$$sgn(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{otherwise} \end{cases}$$

Show it is Turing computable.

Exercise 45.11. Show that if $A \subseteq \omega$ is Turing computable then so is $\omega \setminus A$. Show that if A and B are Turing computable so is $A \cap B$ and $A \cup B$.

Exercise 45.12. Suppose g(x) and h(x) are Turing computable and A is a Turing computable set. Show that f is Turing computable where:

$$f(x) = \begin{cases} g(x) & \text{if } x \in A \\ h(x) & \text{if } x \notin A \end{cases}$$

Exercise 45.13. Show that the set of even numbers is Turing computable. Show that the set of primes is Turing computable.

Exercise 45.14. Show that $e(x,y) = x^y$ is Turing computable. Show that $f(x) = x!$ is Turing computable.

Exercise 45.15. Suppose that $h(z)$ and $g(x,y,z)$ are Turing computable. Define f by recursion, $f(0,z) = h(z)$ and $f(n+1,z) = g(n,z,f(n,z))$. Show that f is Turing computable.

Exercise 45.16. Prove that the set of partial Turing computable functions is the same as the set of partial recursive functions.

Exercise 45.17. Prove that the halting problem for nonwritting Turing machines is decidable. A Turing machine m is nonwritting iff whenever $m(s,a) = (s', a', d)$ then $a = a'$. It is decidable whether or not m halts when given input $x =< x_1, \ldots, x_n >\in A^n$ started on x_1 in state s_0.

Church-Turing Thesis

Here is an excerpt in support of Church's Thesis from Alan M. Turing[2]. Note that Turing uses the word computer for the person that is performing some effective procedure.

" Computing is normally done writing certain symbols on paper. We may suppose this is divided into squares like a child's arithmetic book. In elementary arithmetic the two-dimensional character of the paper is sometimes used. But such a use is always avoidable, and I think that it will be agreed that the two-dimensional character of paper is no essential of computation. I assume then that the computation is carried out on one-dimensional paper, i.e. on a tape divided into squares. I shall also suppose that the number of symbols which may be printed is finite. If we were to allow an infinity of symbols, then there would be symbols differing to an arbitrarily small extent. The effect of this restriction of the number of symbols is not very serious. It is always possible to use sequences of symbols in the place of single symbols. Thus an Arabic numeral 17 or 9999999999999999999 is normally treated as a single symbol. Similarly in any European language words are treated as single symbols (Chinese, however, attempts to have an infinity of symbols). The differences from our point of view between the single and compound symbols is that the compound symbols, if they are too lengthy, cannot be observed at one glance. This is in accordance with experience. We cannot tell at one glance whether 9999999999999999999999999 and 99999999999999999999999999 are the same.

" The behavior of the computer at any moment is determined by the symbols which he is observing, and his 'state of mine' at that moment. We may suppose that there is a bound B to the number of symbols or squares which the computer can observe at one moment. If he wishes to observe more, he must use successive observations. We will also suppose that the number of states of mind which need be taken into account is finite. The reasons for this are of the same character as those which restrict the number

[2] "On computable numbers, with an application to the Entscheidungsproblem", Proceedings of the London Mathematical Society, 2-32(1936), 230-265.

of symbols. If we admitted an infinity of states of mind, some of them will be 'arbitrarily close' and will be confused. Again, the restriction is not one which seriously affects computation, since the use of more complicated states of mind can be avoided by writing more symbols on the tape.

" Let us imagine the operations performed by the computer to be split up into 'simple operations' which are so elementary that it is not easy to imagine them further divided. Every such operation consists of some change of the physical system consisting of the computer and his tape. We know the state of the system if we know the sequence of symbols on the tape, which of these are observed by the computer (possibly with a special order), and the state of mind of the computer. We may suppose that in a simple operation not more than one symbol is altered. Any other changes can be split up into simple changes of this kind. The situation in regard to squares whose symbols may be altered in this way is the same as in regard to the observed squares. We may, therefore, without loss of generality, assume that the squares whose symbols are changed are always 'observed' squares.

" Besides these changes of symbols, the simple operations must include changes of distribution of observed squares. The new observed squares must be immediately recognizable by the computer. I think it is reasonable to suppose that they can only be squares whose distance from the closest of the immediately previously observed squares does not exceed a certain fixed amount....

" The operation actually performed is determined, as has been suggested above, by the state of mind of the computer and the observed symbols. In particular, they determine the state of mind of the computer after the operation. "

Universal Turing Machine

In his paper Turing also proved the following remarkable theorem.

Theorem 45.18 *There is a partial Turing computable function $f(n, m)$ such that for every partial Turing computable function $g(m)$ there is an n such that for every m, $f(n, m) = g(m)$. Equality here means either both sides are defined and equal or both sides are undefined.*

Proof
Given the integer n we first decode it as a sequence of integers by taking its prime factorization, $n = 2^{k_1}3^{k_2}\cdots p_m^{k_m}$ (p_m is the m^{th} prime number). Then

we regard each integer k_j as some character on the typewriter (if k_j too big we ignore it). If the message coded by n is a straight forward description of a Turing machine, then we carry out the computation this machine would do when presented with input m. If this simulated computation halts with output k, then we halt with output k. If it doesn't halt, then neither does our simulation. If n does not in a straight forward way code the description of a Turing machine, then we pretend its coding the empty function, i.e. we just never halt.
QED

46 Trees, Konig's Lemma, Low basis

Definition 46.1 *Recall:*

1. *A nonempty $T \subseteq 2^{<\omega}$ is a tree iff $\sigma \subseteq \tau \in T$ implies $\sigma \in T$.*
2. *For $T \subseteq 2^{<\omega}$ a tree, define:*

$$[T] = \{b \in 2^\omega : \forall n \ \ b \restriction n \in T\}$$

   the infinite branches of T.

3. *For $\sigma \in T$ define:*

$$T(\sigma) = \{\rho \in T : \rho \subseteq \sigma \text{ or } \sigma \subseteq \rho\}.$$

Lemma 46.2 *(Konig's Lemma) If $T \subseteq 2^{<\omega}$ is an infinite tree, then $[T]$ is nonempty.*

Proof
Construct $b \restriction n$ by induction so that $T(b \restriction n)$ is infinite.
QED

Example 46.3 *There exists an infinite recursive tree $T \subseteq 2^{<\omega}$ with no recursive branch.*

Proof
Let K_0 and K_1 be disjoint recursively inseparable sets. Put $\sigma \in T$ iff for all $n < |\sigma|$ and $i = 0, 1$ if $n \in K_{i,|\sigma|}$ then $\sigma(n) = i$. Then the infinite branches of T are the characteristic functions of separating sets.
QED

Proposition 46.4 *Suppose $T \subseteq 2^{<\omega}$ is a recursive tree, and $[T]$ is countable, then there exists a recursive b in $[T]$.*

Proof
There must be a $\sigma \in T$ such that $[T(\sigma)]$ has exactly one element, otherwise T contains a perfect tree. This one element b must be recursive. To see this, note that if $\sigma \subseteq \tau$ and $\tau \not\subseteq b$, then the tree $T(\tau)$ is finite. Hence to determine $b\restriction n$ for any $n > |si|$ we search for an $m > n$ such that exactly one $\tau \in 2^n \cap T(\sigma)$ has an extension at level m.
QED

Proposition 46.5 *(Low basis, Jocusch and Soare) If $T \subseteq 2^{<\omega}$ is an infinite recursive tree, then there exists $b \in [T]$ with $b' \equiv_T 0'$.*

Proof
Inductively construct recursive trees T_e with $T_0 = T$ as follows: Given T_e define the tree:

$$\hat{T}_e = \{\sigma \in T_e : \{e\}^{\sigma}_{|\sigma|}(e) \uparrow\}.$$

Case 1. $\hat{T}_e$ is infinite. Put $T_{e+1} = \hat{T}_e$.

Case 2. $\hat{T}_e$ is finite. Put $T_{e+1} = T_e$.

Since $T_{e+1} \subseteq T_e$ are all infinite trees, the set $\bigcap_e T_e$ is an infinite tree. This is because the intersection of trees is always a tree. It is infinite because for any $n < \omega$ there must be $\sigma \in 2^n$ which is in infinitely many T_e and hence all. By Konig's Lemma there exists $b \in \bigcap_e [T_e]$.

To see, that $b' = 0'$, note that $e \in b'$ iff Case 2 occurred at step e in the construction. But this can be answered uniformly by $0'$.
QED

Exercise 46.6. Find a recursive tree $T \subseteq \omega^{<\omega}$ which is binary branching, and such that $[T] = \{b\}$ where $b \equiv_T 0'$. Binary bran

Exercise 46.7. Prove there exists an infinite recursive subtree $T \subseteq \omega^{<\omega}$ such that T does not contain an infinite recursive chain or an infinite recursive antichain.

T is a subtree of $\omega^{<\omega}$ means that $\sigma \subseteq \tau \in T$ implies $\sigma \in T$ for every $\sigma, \tau \in \omega^{<\omega}$.

$C \subseteq T$ is a chain iff $\sigma \subseteq \tau$ or $\tau \subseteq \sigma$ for every $\sigma, \tau \in C$.

$A \subseteq T$ is an antichain iff $\sigma \subseteq \tau$ for $\sigma, \tau \in T$ just in case $\sigma = \tau$.